A Seminar on Compressing SQL Workloads

Amit A. Pathak

February 16, 2006

Roadmap...

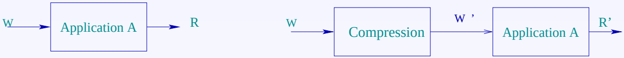
- Introduction.
- Architecture Of Solution
- Search Methods.
- Distance Function.
- Experimental Results.
- Conclusion

Motivation

- Workload - A set of SQL Statements?
- Why it is needed?
  - Index Selection.
  - View Selection.
  - Improving Query Optimization, etc
- Large Size.
- Limits scalability Of Applications.

Workload Compression

- Performance Of Application Should not degreed using compressed workload.
- It is application specific.
- Workload compression should be accomplished efficiently.



|W'| < |W|

FA(W,R) ~ FA(W,R')

Assumptions

- Compressed workload is subset of original workload
- Loss in quality of application due to workload compression can be captured by pair-wise ?Distance? function.

Measuring Quality Of Workload Compression Technique

- $F_A(W', R)$ Measure the result of application A w.r.t W' - Expensive to calculate.
- Use distance function $Distance_A(q_i, q_j)$

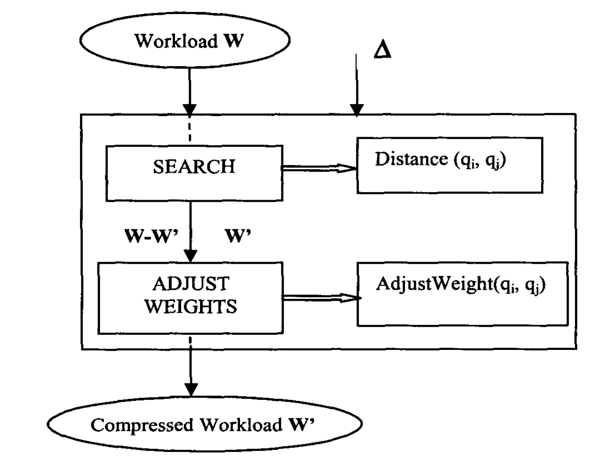
$Distance_A(q_i, q_j)$ measures the loss in quality for statement q_i if q_i is discarded but q_j is present in the workload.

Definition Of Workload Compression Problem

- Given $\triangle$ to be maximum allowable loss in quality and w_i weight of statement q_i.
- Find the smallest workload $W' \subseteq W$ such that for statement $q_i \epsilon W$ not included in compressed workload W' such that,

$$\sum_{q_i} (min_{q_j} w_i * Distance(q_i, q_j)) < \Delta$$
$$\text{Where } q_j \epsilon W'$$

Architecture Of Solution



K-Medoid Algorithm

1. Choose k cluster centers randomly
2. Assign remaining points to closest cluster center
3. Re-compute median for each cluster
4. Repeat steps 2 and 3 until convergence

Search Using K-Medoid Algorithm

1. $Min_k = 0$, $Max_k = |W|$, $W' = W$
2. while $Min_k < Max_k$
3. $k = (Min_k + Max_k)/2$
4. $WTemp = KMedoid(W, k)$ Compute Workload Of size k.
5. Let D be the weighed sum of the distances for each statement in W with closest statement in WTemp.
6. If $(D < \delta)$ $W' = WTemp, Max_k = k - 1$ - OK Reduce the size of workload
7. Else $Min_k = k + 1$ - No increase the workload size.
8. EndIf
9. End While

Search Using All Pair Greedy Algorithm

Maintain Three sets of statements:

1. Keep.
2. Prune.
3. Candidate.

Distance Function

Requirements Of Distance Function:

1. Efficient Calculation.
2. Accurate.

Guidelines for distance function:

1. Partition the workload.
2. Use core information of statement that can be derived efficiently.

Example For Index Selection Problem

Type Of Column	Dist.Func.	Category
Selection Column	Dist-Sel	Reduce no. of rows to be scanned
Group by Column	Dist-GB	Reduce cost of sorting on some operator
Order by Column	Dist-OB	Reduce cost of sorting on some operator
Required Column	Dist-Reqd	Elliminate need to access table altogether

Calculating Dist-Sel

If q_1 is pruned then index $C2$ will get selected.

$$Dist - sel(q_1, q_2) = (.3 - .1) * Cost(q_1,)$$

Selectivity Matrix :

	C1	C2	C3
q1	.1	.3	.2
q2	.5	.2	.4

Weight Adjustments

- If q_i is pruned and its nearest statement is q_j then $w_j = w_j + w_i * \alpha_{ij} / alpha_{jj}$
- α_{ij} is the benefit q_i gets from indices of q_j

Experimental Results

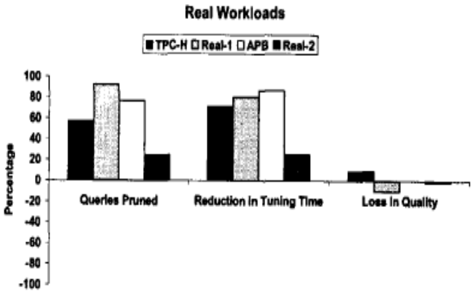
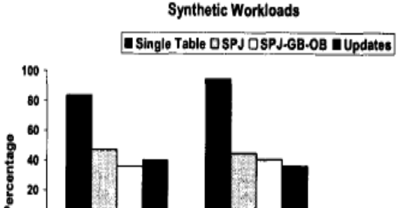


Figure 7. Results on Real and Benchmark Workloads



Experimental Results

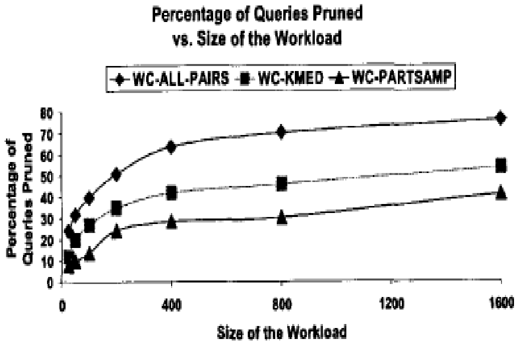


Figure 9. Search Algorithms: Compression Achieved

Experimental Results

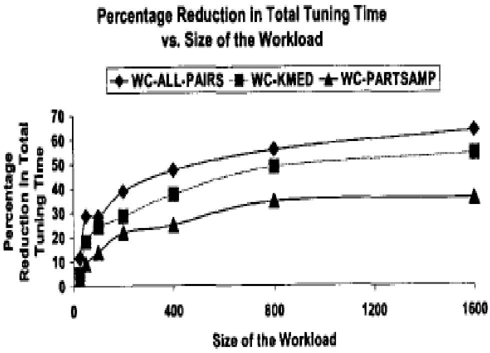


Figure 10. Search Algorithms: Total Reduction in Tuning Time

Experimental Results

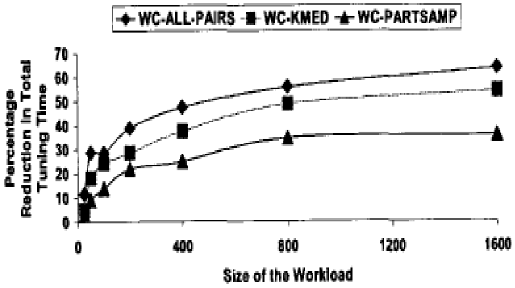


Figure 10. Search Algorithms: Total Reduction in Tuning Time

Number of Distance Computations
vs. Size of the Workload

Experimental Results

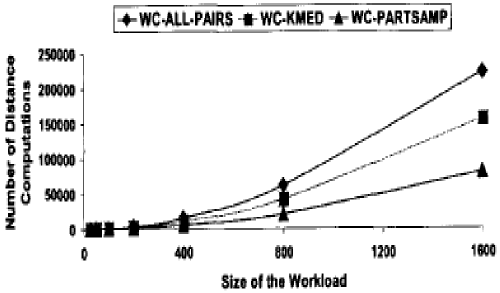


Figure 11. Search Algorithms: Number of Distance Computations

Conclusion

- Provide general platform for workload Compression.
- Solution applicable to a variety of workload-driven applications.
- Provide guidelines for computing distance function.

References I

Vivek Narasayya Surajit Chaudhuri, Ashish Kumar Gupta.
Compressing sql workload.
2002.