CS848 Paper Submission Site

Go To CS848 Site

Submit or Update A Review For Paper #26

Go To CS848 Submissions

It is currently Tuesday 02nd of November 2004 06:05:22 PM EST

Paper # 26 (Download paper of type application/pdf, 291776 bytes)	
Title:	Supporting Top-k Join Quereis in Relational Databases
Abstract:	

You have already finalized your review for this paper.
You can no longer modify it, but you may view it.

If you made a mistake in your review and you want it "unfinalized",
you may send mail to the program chair asking them to unfinalize it

Send yourself this review by email

Attribute	Value
Are you finished with this review?	Finalize, I am done editing
Provide a short summary of the paper	This paper presents a method of performing top-k join queries using the ripple join algorithm. The authors begin with some introductory remarks that motivate the problem and provide some background as to the challenges in performing the rankings. Next, previous methods proposed to solve the top-k join query problem are presented: Chang and Hwang, NRA, TA, etc. Ending their introductory material, the authors present an overview of the ripple join algorithms. The authors next present their new algorithm: the rank join algorithm. The algorithm retrieves tuples, joins them with others already seen and adds them to a set. The algorithm halts when the set contains the k top ranked results and returns them. The authors prove that their algorithm provides the top k results every time, as well as a bounding for the size of the buffer. The authors discuss the effect that the join type has on performance of their algorithm and conclude that joins that reduce the threshold value fastest are optimal for their implementation. The authors next present the use of hash ripple join in performing the rank join algorithm. Thus, the approach is termed hash rank join. Also mentioned and briefly discussed is adaptive join strategies. Although block ripple join is mentioned as applicable, the reader was unable to find a discussion of its use. The authors present a method to switch inputs when it is beneficial to reducing the threshold value. Performance evaluations are provided that compare the new algorithms to the J* rank join algorithm. The new algorithms were found to dramatically reduce computation time when join selectivity is low and the number of requested results is high. The authors finish their paper with some closing remarks.
What is the strength of the paper? (1-3 sentences)	This article presents a novel way of performing rank joins in order to return top-k result sets in an efficient manner. It is very well written and easy to understand. The performance evaluation compares the new algorithms to an established algorithm.

What is the weakness of the paper? (1-3 sentences)	This article makes several assumptions that might not necessarily be valid. Also, some performance evaluation data is missing.
Your qualifications to review this paper	I know the material, but am not an expert
Writing Quality	Excellent
Relevance to query processing?	The paper is relevant to query processing
Experimental Methodology	**Good**
Novelty of paper	This is a new contribution to an established area
Overall paper merit	**A novel or new contribution to this area with good methodology, or an incremental contribution paper that has excellent methodology. A must read for anyone in the area.**
In your opinion, will this paper be important over time?	Good
Provide additional detailed comments to the author	I believe you have presented quite a novel idea. Your writing style is at a level appropriate for readers of all levels and contains very few spelling and grammar mistakes. However, I have two major problems with your presentation: -You assume inputs will be sorted on rank or have an index by rank. Is this a valid assumption? -How does your algorithm perform when the selectivity rises even higher? Is a maximum selectivity of 0.02 really acceptable to experiment with? Overall, I believe this is a great paper.
Additional comments to PC (not seen by author)	**This paper presents a very novel method to solve the top-k join query problem. I do not believe the two minor problems presented to the authors are sufficient grounds to reject the paper.**

Goto Main Index Close Window