CS848 Paper Submission Site

Go To CS848 Site

Submit or Update A Review For Paper #8

Go To CS848 Submissions

It is currently Wednesday 13th of October 2004 09:53:10 PM EDT

Your review has been finalized!

Paper # 8 (Download paper of type application/pdf, 1023552 bytes)	
Title:	Eddies: Continuously Adaptive Query Processing
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 outlines a new relational operator known as an eddy. An eddy is capable of reordering pipelined operators on tuples during execution. The authors begin the paper with a brief overview of some of the environments that the Telegraph system (the system implementing the eddy operator) must adapt to. For example, environments with constantly changing hardware and workload, data and user interface complexities. Attention then turns to the types of system parameters that can change at runtime (i.e. operator costs, data selectivities and input data arrival rates), and architectural assumptions on which Telegraph is based. Next, the authors turn their attention a reason for reordering query operators. One reason is a synchronization barrier, where one input to a join takes significantly longer to produce input tuples than another. In this case, the authors propose using moments of symmetry (where a complete phase of computation has finished) to swap inputs to produce a faster join. This works for nested-loop joins, but not necessarily for index joins. The authors comment that they favour join algorithms that have frequent moments of symmetry so that they may change the order of inputs on the fly more often to optimize the join as system parameters change. One family of join algorithms experiencing a high frequency of moments of symmetry is the ripple join family of algorithms. The authors present several other families and reasons why they are not very useful in Telegraph. The authors next turn their attention to a discussion of the implementation of eddies in the Telegraph system. Telegraph was built upon a workload-aware system called River. River provides near-record performance on tasks because of its scaling abilities during periods of fluctuating workload. Telegraph has a preoptimizer that decides on the initial

	assignment of inputs to operators. This problem is solved by finding a minimum spanning tree for assignment of inputs to join operators in a query tree. Tuples then enter the eddy operator, are sent to operators that are ready to process it, and returned to the user as output once they have been processed by all operators. This is facilitated with "ready" and "done" vectors. A ready bit for an operator is turned on if that tuple can be processed by that operator. The done bit for an operator is turned on if that tuple has been processed. Tuples with all done bits are returned. The following section presents the routing algorithm for tuples within an eddy. An eddy's tuple buffer is a priority queue that assigns high priority to previously manipulated tuples, and low priority to new input tuples. In this manner, old tuples are completely processed before new tuples are started. The authors present a lengthy discussion of experimental results. As the paper draws to a close, the authors relate their work to the work of others. Future directions for research are given as well as a short conclusion.
What is the strength of the paper? (1-3 sentences)	This paper presents a new way to optimize queries: reordering of operations on-the-fly. It does a fairly good job explaining the underlying concepts necessary to implement such a system.
What is the weakness of the paper? (1-3 sentences)	There does not appear to be any smooth transition between discussed topics in this paper (i.e. the paper does not flow well). Topics are simply discussed and left without relation to other portions of the paper. As well, I am not convinced that the experiments performed demonstrate how eddy's behave with real world parameters and variances.
Your qualifications to review this paper	I know the material, but am not an expert
Writing Quality	Average
Relevance to query processing?	The paper is relevant to query processing
Experimental Methodology	**Poor**
Novelty of paper	This is a new contribution to an established area
Overall paper merit	**The paper is a novel or new contribution with average/weak methodology, or an incremental contribution that has good methodology. Someone in the area should read it**
In your opinion, will this paper be important over time?	Average
Provide additional detailed comments to the author	This paper presents a new contribution to an established area. Although I believe you have presented a fantastic concept, I have two serious issues with your paper: -I am not convinced that your performance analysis is a true representation of real world conditions. Why not just run the system on actual data instead of introducing delay units and spin loops, etc.? Did you attempt to acquire a large data set for testing purposes? Why not generate your own sets of data? -Your writing organization is good, but I don't have a sense that this paper flows from one topic to another. One topic is presented, then another, etc. There is no segue or connection between most topics. I would suggest that you fix these deficiencies and resubmit your paper for consideration.
Additional comments to PC	**Although this paper presents a fascinating concept, I do not believe it is publishable in its current state. At this point, I believe the deficiencies outweigh the novelty, and**

(not seen by author)	**this paper should not be published. I would encourage the authors to fix the problems associated with their work and resubmit their paper for consideration.**

Goto Main Index Close Window