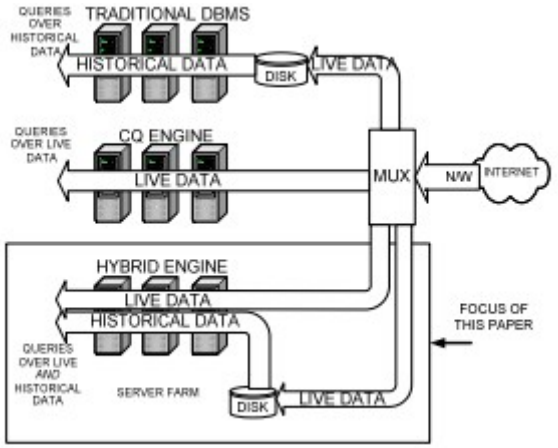
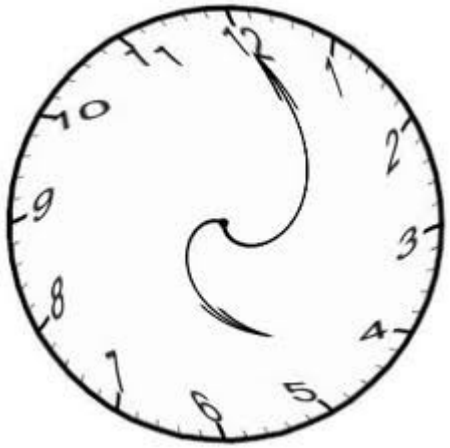
Remembrance of Streams Past : Discussant



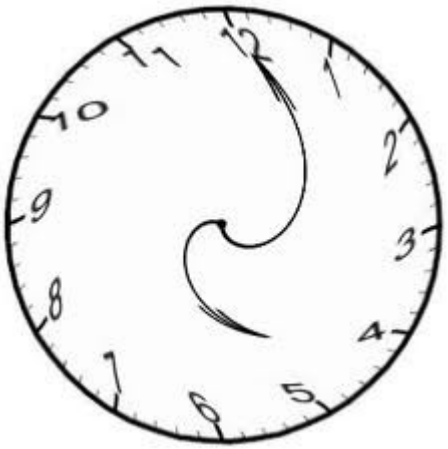
By Andrew Osgood

Sirish Chandrasekaran
EECS Department, UC Berkeley
sirish@cs.berkeley.edu

Michael Franklin
EECS Department, UC Berkeley
franklin@cs.berkeley.edu



Back to 2004



- “End of an Architectural Era” published 2007*
- No mention of IM Database Systems
- Only examining the interaction between disk and stream

* Michael Stonebraker et al. The end of an architectural era: (it's time for a complete rewrite), Proceedings of the 33rd international conference on Very large data bases, September 23-27, 2007, Vienna, Austria

The Problem ?

- Overloaded disks prevent queries from interacting with live streams and historical data quicky

The Solution ?

- Reduce I/Os with Data Reduction Techniques and a new System : OSCAR

OSCAR

Overload-sensitive Stream Capture and Archive Reduction

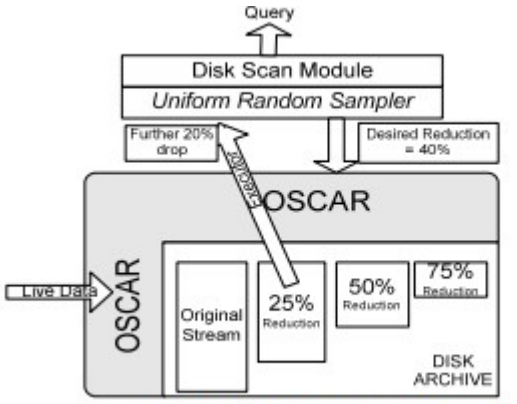


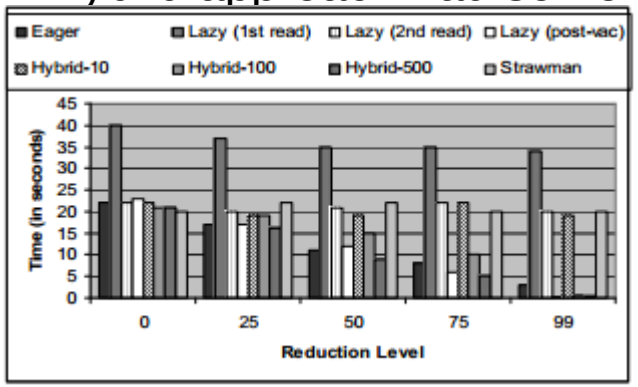
Figure 3: 100 ft. view of solution

Used to reduce disk I/Os

- Only finite partitions (user wants 40 % versus the reduction of 50%)
- Necessitates user defined functions for reduction
- Different overload plans for different DR schemes

Eager, Lazy, and Hybrid

- Each possess their own tradeoffs
- Eager suffers on writes, Lazy suffers on reads
- Hybrid: best of both worlds
- Each examined analytically and experimentally
- Hybrid approach had some interesting results...



Little discussion of 1st read's bottleneck for the Lazy approach

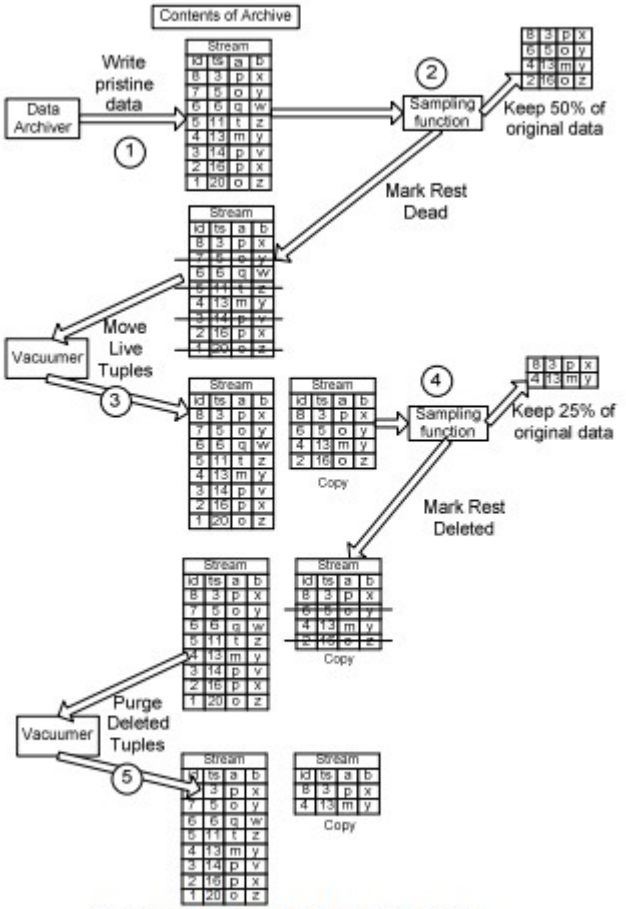


Figure 7: OnReadModify in action

The RandomizeThenSort Approach

Randomize ThenSort	$B(1+\frac{I}{R*S})$	$\sum_{j=1}^{B/R}\lceil R * f'(j)\rceil$

Analytically it checked out

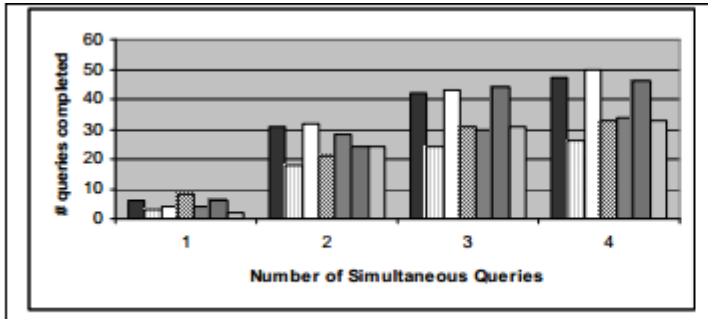


Figure 15: #queries completed within 20M tuples insertion, 25% reduction

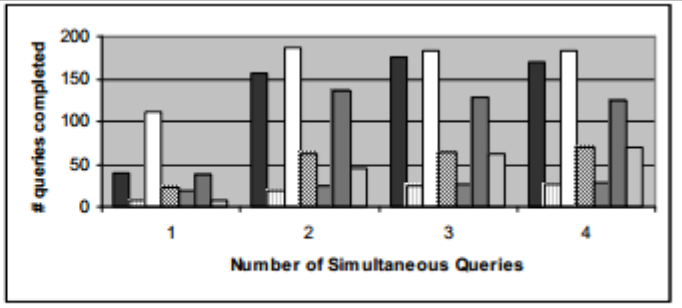


Figure 16: #queries completed within 20M tuples insertion, 75% reduction

Experimentally, it fell short

Why ?

- Problems with the OS *
  - “When the OS guesses that a query is not performing sequential I/O (in our system this happens if two blocks that are accessed consecutively are more than 8 blocks apart), it gets scheduled less often.”
  - “the behavior of the file-system prevents us from directly proving our central hypothesis”
  - “the OS process scheduler cause our hybrid OSCAR design to show widely different behavior”

*S. Chandrasekaran and M.J. Franklin, "Remembrance of Streams Past: Overload-Sensitive Management of Archived Streams," Proc. Int',l Conf. Very Large Data Bases (VLDB), 2004.

Results ?

- Is this the OSCAR we're dealing with ?

- No, experimental results are mixed
- Inherently not a bad idea, more work to be done

Questions ?

Text Source : S. Chandrasekaran and M.J. Franklin, "Remembrance of Streams Past: Overload-Sensitive Management of Archived Streams," Proc. Int',l Conf. Very Large Data Bases (VLDB), 2004.

Images either from : S. Chandrasekaran and M.J. Franklin, "Remembrance of Streams Past: Overload-Sensitive Management of Archived Streams," Proc. Int',l Conf. Very Large Data Bases (VLDB), 2004.

Or images.google.com