The Skyline Operator

Stephan Borzsonyi, Donald Kossmann and Konrad Stocker

Abhishek Srivastava
Student ID: 861307778

February 21, 2017
CS 236, Winter 2017

The problem:
The paper discusses the lack of operation currently present in the existing RDBMS relations to find an optimal queries based on different attributes with different properties.

The contribution:
The authors propose skyline operator to solve the issues described above. From large set of spatial data points a set of interesting points are extracted using this operation. These selected points must show the property of not being dominated by other points which means with respect to other points it is better in at least one dimension and is comparably good or same in other dimensions. Skyline operator just tries to maximize the scoring function for set of points among all points.

The method:
The Authors approach to implement new logical skyline operator is to extend existing database system and make the integration as simple as possible. Author presented three main variants to implement with multiple ways among each one of them.

Skyline operator implementation variants:

- **Sorting**: Skyline is calculated by applying sort on the data and comparing tuples with the predecessor tuples. If data does not fit in main memory sort-merge variant can be applied but it will lead to very high I/O cost. This method works for data with dimensions less than 3.

- **Block Nested Loops**: Most basic way is to apply nested loop over whole data and compare each tuple to compute skyline. But this is extend by keeping incomparable tuples window in main memory and comparing with other tuples and deciding whether to keep or throw based on whether the tuple dominates or is incomparable. Other variants of this methods are **Maintaining self organizing window** and **Replacing tuples in window based on their dominance**.

- **Divide and Conquer**: It works by diving the whole data sets in 2 partitions based on their dominance, choosing skylines from both partitions and merging them to get skyline points of overall data. The shortcoming of this can be that whole data cannot be stored in main memory. Its other variants solve this issue **M-way partitiong** which divides the whole data in M partitions and then merge them. **Early Skyline** which tries to find skyline point before dividing and merging.

Author also presented how these operations can be applied using B-trees and R-trees.

Comments:
The paper presents a new operation called skyline. Authors also presents the performance of their multiple variants of skyline operations over multi-dimensional data, with data also being generated using different distributions. They are also compared with respect to the size of buffer and database. Author concludes that for good cases Block-Nested should be used and for rough cases Divide and conquer method should be applied.

However, Some of the short comings were:

- No perfect methods to determine the size of skyline set.

- How the operation behaves in parallel database systems.