CAS CS 561: Data Systems Architectures
Data-intensive Systems and Computing Lab
Department of Computer Science
College of Arts and Sciences, Boston University
http://bu-disc.github.io/CS561/

CS561 Spring 2021 - Research Project

Title: Designing sortedness-aware access methods

Background: Real-time analysis of data is becoming more popular and a crucial requirement for many business operations, which includes IoT data analytics, social data management, and disaster control. In many cases, this data comes with *implicit ordering*, i.e., the incoming data set comes implicitly partitioned for some attributes which are correlated with time. This implicit ordering of data may cause algorithms to perform sub-optimally in several cases. For example, if the data comes in sorted order they trigger the worst-case runtime ($O(n^2)$) for a quick sort routine. Similarly, the insert cost for new entries may vary from O(log n) (for completely scrambled datasets) to O(1) (for fully sorted datasets), depending on the *sortedness* of the datasets and the access methods used to insert the data. Prior knowledge about the data, in such a case, is helpful in tuning the performance in an optimal way. Such analogies are also applicable to real-life data storage systems and access methods, which deal with variable workload characteristics, while being agnostic of any implicit structure present within the workload.

Objective: This project aims to design new access methods, which can exploit any implicit ordering or structure in a workload. The proposed access method should (i) ensure competitive performance as compared to state-of-the-art when subject to benchmark workloads and (ii) exploit any implicit ordering, if present, in the workload to offer better performance than that offered by the state-of-the-art. The following workflow is to be adhered in course of this project.

(a) Review the existing metrics that are proposed to quantify the orderliness or pre-sortedness of a dataset
(b) Design a metric (or justifiably choose one from the existing ones) that characterizes the orderliness of data, given a workload
(c) Design access methods that have properties (i) and (ii)

[1] Manos Athanassoulis, Anastasia Ailamaki. BF-Tree: Approximate Tree Indexing. PVLDB 7(14): 1881-1892 (2014)
[2] Heikki Mannila. Measures of Presortedness and Optimal Sorting Algorithms. IEEE Trans. Computers 34(4): 318-325 (1985)