UNIVERSITY OF AMSTERDAM

Space Efficient Indexes for the Big Data Era

E. Sidirourgos

Summary

The *big data* era is characterized by the challenges put forward by both data-intensive scientific discovery and the e-commerce surge. Scientific discovery has shifted from being an exercise of theory and computation, to become the exploration of an ocean of observational data. State-of-the-art astronomical observatories and modern scientific instruments produce every day petabytes of information. Moreover, e-commerce sales have grown in the last decade and expected to account for more than 10% of retail sales in the near future. Analyzing user generated data and web logs is crucial for increasing competitiveness, creating targeted advertisement and advanced recommendation systems, and providing quality of service. Enterprises gather huge amounts of data, to be continuously queried and updated for fast user experience, but also batch analyzed over long periods to design business plans and economical strategies. The big data challenges in the size of the data collections, the speed of updates, and the diversity of the data models are summarized in the so called "3Vs" *volume, velocity, variety*.

The predominant answer to the big data challenge, given by the data management community, is the raw power of big data center installations, complemented by new technologies focusing on scalable distribution of data and operations, such as MapReduce and Hadoop. In addition, system designers have build database warehouses to work on top of these distributed environments, such as Hive, Pig, Impala, and more. These systems are spread across multiple machines and therefor should be easy to deploy and initialize. Moreover, accessing local partitioned data remains a bottleneck, thus there is still a clear need for *space efficient indexes* that are lightweight to build and maintain. Such space efficient indexes consume only sub-linear to the indexed data space, are fast to create (usually a single scan), and also are easy to update (linear to the size of the appended data). In addition, usually they are secondary structures, meaning that they do not dictate the placement of the indexed data, thus not requiring expensive read and write steps during creating or updates.

In this thesis we present four space efficient indexes that satisfy the above requirements. Each one of them is designed to address the requirements of a specific applica-

tion. First, we present *column imprints*, a cache conscious secondary index for column stores capable of answering range queries fast. Imprints are particularly handy for numerical domains in scientific databases with a large number of attributes per table. We next introduce a new variant of Bloom filters called *split Bloom filters*, designed to restrict access to cold stores in the presence of skew access. The applicability of such index is in large e-commerce sites that keep in memory hot sets of users and products, while old and outdated data are stored in slower memories. We then introduce a type independent index that produces an order preserving hash function. It is used to index large XML repositories. The main challenge to overcome is that XML elements are typeless and predicates type agnostic, e.g., a predicate matches both strings and numerical values. Finally, we present a space efficient string index based on grams, called *qs-grams*. This index is capable of answering sub-string queries on huge collections of BLOB's or documents, faster than the state-of-the-art *n-grams* and by using a 1-1 ratio of storage. *qs-grams* are designed for pattern matching applications, such as genome sequence alignment or detecting malicious snippets of binary code on disks.