Database Kernels Tailored for Data Exploration

Stratos Idreos
Harvard University

Abstract

Data exploration is about efficiently extracting knowledge from data even if we do not know exactly what we are looking for. In this tutorial, we survey recent developments in the emerging area of database systems tailored for data exploration. Specifically, we survey recent literature that focuses on the vision of database kernels that are tailored for data exploration by adapting their storage and access patterns to varying workloads.

Description

Traditional data management systems assume that when users pose a query a) they have good knowledge of the schema, meaning and contents of the database and b) they are certain that this particular query is the one they wanted to pose. In short, we assume that users know what they are looking for. In addition, traditional DBMSs are designed for static scenarios with numerous assumptions about the workload. For example, state-of-the-art systems assume that there will be a tuning phase where a database administrator tunes the system for the expected workload. Again, this assumes that we know the workload, we know that it will be stable and we also have enough idle time and resources to devote to tuning.

The above assumptions were valid for the static applications of the past and they are still valid for numerous applications today. However, as we create and collect increasing amounts of data, we are building more dynamic and data-driven applications that do not always have the same requirements that database systems have tried to address during the past five decades. This is true across many businesses as well as scientific research that continuously becomes more data-driven.

In this tutorial we survey recent literature on designing from the ground up new database kernels that are tailored to provide efficient access to data even if we do not know what we are looking for up front, i.e., systems that are tailored for data exploration [1, 2]. Specifically, we will survey recent developments in adaptive indexing [3], adaptive data loading [4], adaptive storage [5] as well as new ideas on how to provide exploration capabilities by co-designing user-interfaces with database kernels [6].

Adaptive Indexing. In a data exploration scenario we are searching for interesting data patterns without knowledge of what we are looking for. Yet traditional database systems rely heavily on tuning actions and accurate workload knowledge to achieve good performance. One of the most critical tuning actions is that of choosing the proper set of indexes. Making strict a priori choices means that a system is not well prepared for an exploratory scenario where users may focus on arbitrary data parts at different times. Research on adaptive indexing introduces the idea of creating indexes incrementally and adaptively during query processing based on the columns, tables and value ranges that queries request. Indexes are built gradually; as more queries arrive indexes are continuously fine-tuned. Adaptive indexing has been studied to improve selections in column-stores, and has been shown to work in late materialization architectures, to allow for incremental and partial projections, to be robust in workload changes, to absorb updates efficiently and adaptively and to enable multi-query processing via concurrency control. In addition, the basic algorithms have been studied in depth in the face of trade-offs such as adaptation speed and initialization costs as well as they have been optimized for modern hardware.

Adaptive Loading. During data exploration not all data is needed. Adaptive loading exploits this fact and introduces the notion that users can start querying a database system (with efficient response times) even before all data is loaded or even leaving some parts of the data unloaded, effectively enabling efficient raw data access.

Adaptive Storage. The way we store data defines the best possible ways to access it. There is no perfect storage layout; instead there is a perfect layout for each individual data access pattern. Modern systems rely on static layouts and build the whole architecture around a single layout. In a data exploration scenario we cannot a priori decide what is a good layout, as we do not know the exact query patterns up front, leading to sub-optimal performance for traditional static systems. In this part of the tutorial, we discuss recent work that aims at removing this problem through adaptive storage.

Gesture-based Data Exploration. Last we will discuss the vision towards systems that make interaction with a data simple and intuitive without requiring significant expertise from data scientists. Users interact with the system via gestures as opposed to complex languages. The user interface is co-designed with the database kernel in order to provide very fast reaction to user input and the illusion of “touching the data”.

Summary. This tutorial takes a close look into database kernel design and discusses designs that do not require any set-up or workload knowledge but still provide efficient response times by being able to adapt to incoming queries and data. The ideas on data exploration kernels will be presented in the context of modern main-memory optimized systems with columnar or hybrid data layouts that fully utilize modern hardware.

Bio

Stratos Idreos is an assistant professor of Computer Science at Harvard University where he leads DASlab, the Data Systems Laboratory@Harvard SEAS. Stratos works on data systems architectures with emphasis on designing systems for big data exploration. For his doctoral work on Database Cracking, Stratos won the 2011 ACM SIGMOD Jim Gray Doctoral Dissertation award and the 2011 ERCIM Cor Baayen award as from the European Research Council on Informatics and Mathematics. In 2010 he was awarded the IBM zEnterpise System Recognition Award by IBM Research, and in 2011 he won the VLDB Challenges and Visions best paper award. In 2015 he received an NSF CAREER award and was awarded the 2015 IEEE TCDE Early Career Award from the IEEE Technical Committee on Data Engineering.

References

1. S. Idreos, O. Papaemmanouil, and S. Chaudhuri. Overview of Data Exploration Techniques. In Proceedings of the ACM SIGMOD International Conference on Management of Data, 2015
2. S. Idreos. Big Data Exploration. Taylor and Francis, 2013
3. S. Idreos, M. L. Kersten, and S. Manegold. Database cracking. In Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR), 2007
4. S. Idreos, I. Alagiannis, R. Johnson, and A. Ailamaki. Here are my Data Files. Here are my Queries. Where are my Results? In Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR), 2011
5. I. Alagiannis, S. Idreos, and A. Ailamaki. H2O: a hands-free adaptive store. In Proceedings of the ACM SIGMOD Conference on Management of Data, 2014
6. S. Idreos and E. Liarou. dbtouch: Analytics at your fingertips. In Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR), 2013