Distribution-Oblivious Databases

Hannes Mühleisen
Database Architectures Group
Centrum Wiskunde & Informatica
Amsterdam, The Netherlands
hannes@cwi.nl

Distributed databases have long been a focus of research attention. There are many reasons for this, but the ability to handle more data or more requests within a given time unit are is the central goal. Also, by combining many small computers into a single large logical database, one can take advantages of scale effects with regards to hardware costs. However, managing distributed systems is not a trivial task. Complex scheduling decisions have for example to be taken to maximize the total throughput of a distributed database. Even more serious is the impact of distributed operations to the design of a database system. Not only does the database have to achieve competitive performance on a single node, but the challenges created through the distributed operations have to be considered as well.

We propose a different approach, where a very thin coordination layer is placed on top of a set of nodes all running a single-node database with access to the same data. This coordination layer manages the assignment of queries to nodes, management of load and is tracking the liveliness of nodes as well. However, this layer also splits up queries into smaller parts, for which less data has to be read from disk. We combine this layer with a distributed file system and a preferential placement of sub-queries reading the same data to the same node in the network. Overall, this approach allows efficient execution of complex analytical queries on a large number of nodes. At the same time, the databases on the nodes remain completely oblivious to the distributed operations. By only relying on standard SQL as a query language for sub-queries, we are also able to compare various single-node databases with regards to their performance in this environment. This approach also represents a novel trade-off between the federated and distributed databases: The former rely on API-level collaboration without insight into the queries, while the latter fully internalize the distributed operations. We take the middle ground, where each database operates locally, with queries being executed in a distributed way.

Conceptually, we view databases as a function that transforms a subset of a sequence of bytes from secondary storage (the data on disk) into a relational query result for each given query. Our hypothesis is that by manipulating the queries we can reduce the number of bytes read to a fraction. It is of course in the utmost interest of every database to limit the amount of data read to answer a query, simply reading the complete data set from disk might be effective, but not efficient. It is this property that allows us to achieve exhibit considerable overall efficiency through the (ab)use of caches in each of the participating nodes. From a systems viewpoint, we argue against re-invention of features that are nowadays common to distributed environments such as distributed file management, coordination or group membership.

In the talk, we will give a preview into this ongoing work. We describe our model, the overall process of query rewriting, sub-query scheduling and execution as well as the recombination of the results into the overall response. Also, we will present first – very promising – results from comparing this system with competing approaches.