Integrating analytics with relational databases

Raasveldt, M.

Summary

The database research community has made tremendous strides in developing powerful database engines that allow for efficient analytical query processing. However, these powerful systems have gone largely unused by analysts and data scientists. This poor adoption is caused primarily by the state of database-client integration: current methods of combining databases with analytical tools are slow and cumbersome. Instead, data scientists have opted to re-invent database systems by developing a zoo of data management alternatives that perform similar tasks to classical database management systems, but have many of the problems that were solved in the database field decades ago.

In this thesis we attempt to overcome this challenge by investigating how we can facilitate efficient and painless integration of analytical tools and relational database management systems. We focus our investigation on the three primary methods for database-client integration: client-server connections, in-database processing and embedding the database inside the client application.

For each of these methods we take an extensive look at implementations in existing systems, and evaluate how they perform in the context of standard analytical workloads. We evaluate the benefits and drawbacks that they exhibit in this context, both in terms of query performance and usability.

We propose several novel techniques that improve upon the state-of-the-art. We demonstrate a new client-server protocol that is optimized for bulk-transfer of large data sets. We showcase our MonetDB/Python UDFs, that improve on large in-database processing efficiency through vectorized execution. We describe MonetDBLite, an embedded version of the MonetDB database system that we have efficiently integrated with Python and R. The techniques that we propose have all been integrated and tested in real database systems, showing that these solutions are not just theoretical but practically applicable as well.

In the final chapter we showcase DuckDB, a new data management system that we have built from scratch. When building DuckDB, we took all the lessons that we learned from developing efficient database-client interfaces and applied them.

In conclusion, the techniques that we have developed enable significantly more efficient and usable integration between database systems and analytical tools. Nevertheless, there is still more to be explored in this area. We close this thesis with a program for future research, as well as sketches for solutions to them.