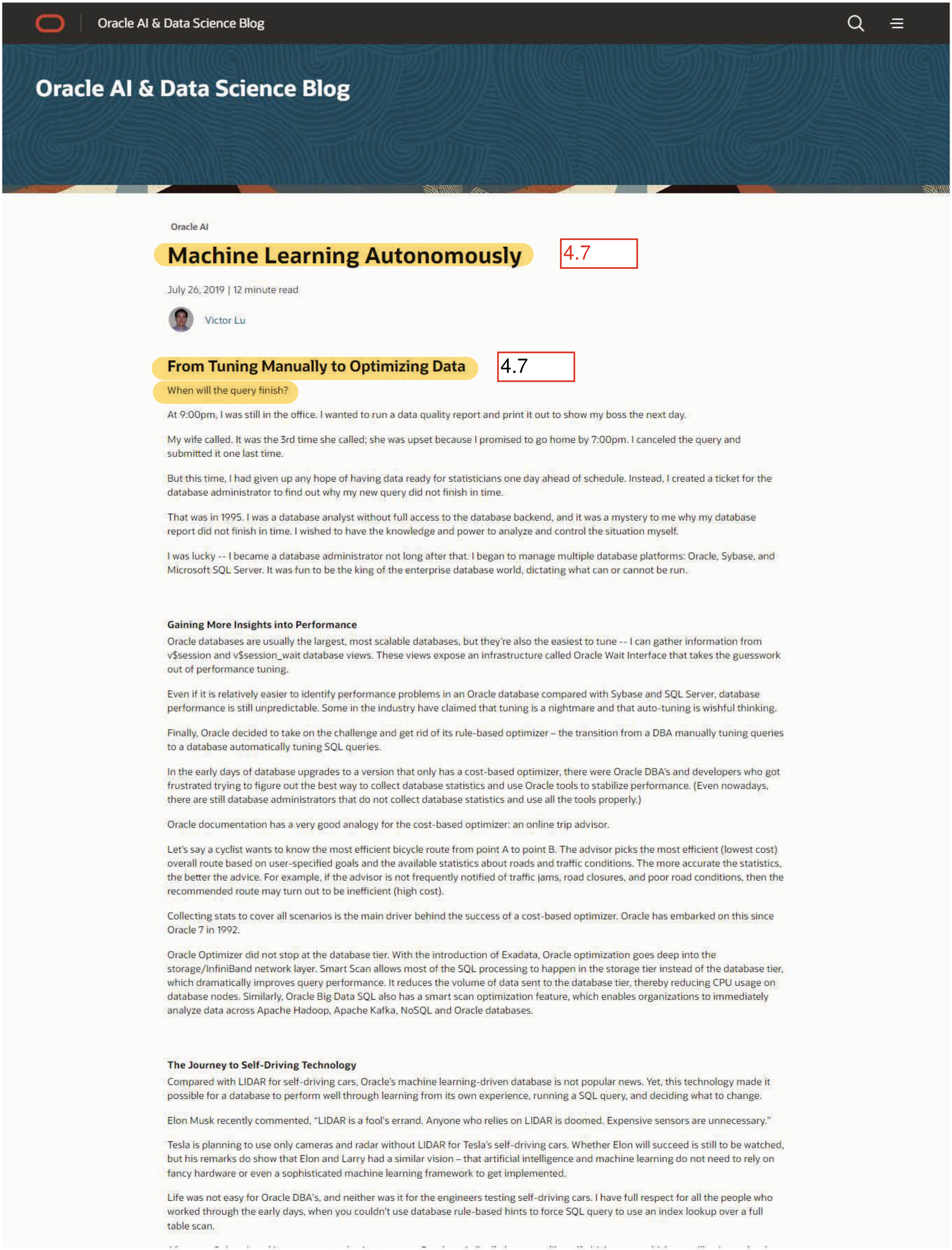

Oracle AI & Data Science Blog

Oracle AI

Machine Learning Autonomously

4.7

July 26, 2019 | 12 minute read

Victor Lu

From Tuning Manually to Optimizing Data

4.7

When will the query finish?

At 9:00pm, I was still in the office. I wanted to run a data quality report and print it out to show my boss the next day.

My wife called. It was the 3rd time she called; she was upset because I promised to go home by 7:00pm. I canceled the query and submitted it one last time.

But this time, I had given up any hope of having data ready for statisticians one day ahead of schedule. Instead, I created a ticket for the database administrator to find out why my new query did not finish in time.

That was in 1995. I was a database analyst without full access to the database backend, and it was a mystery to me why my database report did not finish in time. I wished to have the knowledge and power to analyze and control the situation myself.

I was lucky -- I became a database administrator not long after that. I began to manage multiple database platforms: Oracle, Sybase, and Microsoft SQL Server. It was fun to be the king of the enterprise database world, dictating what can or cannot be run.

Gaining More Insights into Performance

Oracle databases are usually the largest, most scalable databases, but they're also the easiest to tune -- I can gather information from v$session and v$session_wait database views. These views expose an infrastructure called Oracle Wait Interface that takes the guesswork out of performance tuning.

Even if it is relatively easier to identify performance problems in an Oracle database compared with Sybase and SQL Server, database performance is still unpredictable. Some in the industry have claimed that tuning is a nightmare and that auto-tuning is wishful thinking.

Finally, Oracle decided to take on the challenge and get rid of its rule-based optimizer – the transition from a DBA manually tuning queries to a database automatically tuning SQL queries.

In the early days of database upgrades to a version that only has a cost-based optimizer, there were Oracle DBA's and developers who got frustrated trying to figure out the best way to collect database statistics and use Oracle tools to stabilize performance. (Even nowadays, there are still database administrators that do not collect database statistics and use all the tools properly.)

Oracle documentation has a very good analogy for the cost-based optimizer: an online trip advisor.

Let's say a cyclist wants to know the most efficient bicycle route from point A to point B. The advisor picks the most efficient (lowest cost) overall route based on user-specified goals and the available statistics about roads and traffic conditions. The more accurate the statistics, the better the advice. For example, if the advisor is not frequently notified of traffic jams, road closures, and poor road conditions, then the recommended route may turn out to be inefficient (high cost).

Collecting stats to cover all scenarios is the main driver behind the success of a cost-based optimizer. Oracle has embarked on this since Oracle 7 in 1992.

Oracle Optimizer did not stop at the database tier. With the introduction of Exadata, Oracle optimization goes deep into the storage/InfiniBand network layer. Smart Scan allows most of the SQL processing to happen in the storage tier instead of the database tier, which dramatically improves query performance. It reduces the volume of data sent to the database tier, thereby reducing CPU usage on database nodes. Similarly, Oracle Big Data SQL also has a smart scan optimization feature, which enables organizations to immediately analyze data across Apache Hadoop, Apache Kafka, NoSQL and Oracle databases.

The Journey to Self-Driving Technology

Compared with LIDAR for self-driving cars, Oracle's machine learning-driven database is not popular news. Yet, this technology made it possible for a database to perform well through learning from its own experience, running a SQL query, and deciding what to change.

Elon Musk recently commented, "LIDAR is a fool's errand. Anyone who relies on LIDAR is doomed. Expensive sensors are unnecessary."

Tesla is planning to use only cameras and radar without LIDAR for Tesla's self-driving cars. Whether Elon will succeed is still to be watched, but his remarks do show that Elon and Larry had a similar vision – that artificial intelligence and machine learning do not need to rely on fancy hardware or even a sophisticated machine learning framework to get implemented.

Life was not easy for Oracle DBA's, and neither was it for the engineers testing self-driving cars. I have full respect for all the people who worked through the early days, when you couldn't use database rule-based hints to force SQL query to use an index lookup over a full table scan.