Paper Evaluation

NoDB: Efficient Query Execution on Raw Data Files

Michael Lanthier

NoDB describes an efficient method of querying over raw data. NoDB plays tricks when parsing in the data by only transforming values to binary that are required and performing selections to reduce the amount of transformations it must do. It also uses a positional map which keeps track of the location of data and statistics as queries happens. This allows NoDB to efficiently jump around within files and get close enough to the data it needs to read to reduce the amount of time spent parsing. By reading the data in situ and cutting out the database loading step, NoDB is able to perform better than most modern databases when seeing a limited set of queries.

Questions/Comments

1. Is loading a database from scratch a common problem? This is definitely a really cool construction of a system but how often do people have a bunch of data files just sitting around that they need to load in to a DB to run queries on? I could see this being a nice thing for scientific computing as I've seen people have to take tons of data from someone working in excel and exporting it as a CSV and trying to do operations on it.

2. Earlier on they mention that one thing they were trying to avoid was the earlier queries being less efficient or slower than the older queries and they wanted to minimize that. Since the positional map must be generated, NoDB will still suffer a similar fate with the first few queries being slower than queries as time goes on. In testing once it is generated it does work fairly well it seems.

3. How would PostgresRAW handle with more than the nine queries that they serve? If someone plans to keep querying their data over and over and over again would it be worth it to just load it into a PostgreSQL database and let it sit? Especially in their scientific computing example, if sets of queries come with significant gaps in between them (hours or days) as the user analyzes the data it may just be better in the long term to use straight PostgreSQL. At what point does keeping the data in situ become not worth it.