Ikhlas Attarwala
MSAI 339 – Data Science Seminar
10-10-18 – Background Reading

Summarizing "Impala: A modern, open-source SQL engine for Hadoop" by Kornacker, Behm, …

Kornacker and co. wrote a fantastic paper on Cloudera Impala, a new open-sourced MPP SQL query engine designed to integrate the familiarity of traditional DBMSs with the tolerance and versatility of the Hadoop framework by Apache Software. Before I get into their approach, I should mention that I was very impressed by their critical evaluation of Impala and future sights discussed throughout the paper. Now, the authors first ease into the similarity of use and accommodations Impala has to offer, along with the goals of the system. It was impressive how they immediately and carefully transition into a contrast of current approaches, while in the same paragraph state that Impala's performance "is on par or exceeds that of commercial MPP analytic DBMSs". Of course, they follow this up with statistics to show just how much faster of a performance engine it is compared to alternatives. The Impala engine works as a query processing system for Hadoop. Using ODBC or JDBC, SQL queries are sent to Impala where background daemon processes called impalads coordinate the query. This parses the query, analyzes it for semantic information, and optimizes a plan for execution. Using standard components of Hadoop such as HDFS, data is provided to the coordinating impalad which sends outputs and results back to the client. It should be noted that while it does query large volumes of data, it does not replace batch processing, such as by MapReduce.

There were a few features from this paper that I really liked seeing here. The first is how the authors critiqued Impala, sparing no expense at illustrating where and how the system wasn't ideal or had services yet to be implemented. Being able to excitedly unveil a product while discussing the limitations it has really does show a maturity in the authors' handling. Additionally, the authors do well to highlight every integration and service by Impala, such as showing schema for query processing, query optimization, even impact in algorithmic complexity. Lastly, the authors include a "Roadmap", and this is great because they're telling us what is left to be done, and what else we could see in the future of Impala. It's modest but also great for clients to realize that there's potential for growth here.

There are some parts of this paper that don't seem to flow well with the rest. I feel as though the authors don't make a very good point of expressing confidence throughout the paper. There are times in which the authors say they *expect* results to be a certain way in the near-future, or that they are still investigating solutions for problems (ex. resource management with other frameworks). I might have noticed another issue, where as Impala seems to be I/O bound, the paper doesn't go into great detail on how to handle issues such as if a storage device were to encounter failure, query behavior would slow dramatically. It does mention that "there is no built-in failover mechanism in Impala, instead … automatically move to the new process instance" (page 4).

Issues such as the one just mentioned above may be a direction to take this research into the future, to hone in on those smaller issues without solutions. Despite this, Impala seems to be a powerful, fast and as-flexible-as-Hadoop query engine. In truth, the Roadmap section of this paper already suggests future directions to take Impala, and of them, access to remote data storage such as the cloud data storage seems to be one of the smartest choices for now (although this article was from 2014, so it has, likely, already been implemented by now).