Deploying a steered query optimizer in production at Microsoft

Query optimization for complex workloads are hard. They are either sub optimized or trained on generic dataset that does not represent the real workload like TPC-H. Each customer has a specific use case and the query optimizer should be adjusted accordingly which is known as instance optimization. Systems like Neo replace the entire query optimizer to create the plans directly. Bao choices a less aggressive approach of steering the query towards optimality. Microsoft continued this line of work on real world workload in their cloud system SCOPE and made QO-Advisor. There were three major problems: Due to multiple rule hints, it was difficult to identify which rule worked well and which didn't. Second, experimentation is very costly and third, working it on unknown queries due to incorrect cost models. The Scope optimizer uses a combination of data stats and other heuristics and Scope provides A/B testing infrastructure. Also more than 60% of the Scope jobs are recurring. Also there are 256 rules in Scope optimizers. Integrating system such as BAO is non trivial as the search space becomes 2^256. To solve this, QO-Advisor was introduced. It provides rule hints for query plans. It follows the single rule flip and does not provide full rule configuration and Learns over estimates rather than runtimes(I guess good for not overfitting). It currently runs offline once a day and is deployed over different SCOPE clusters. It uses a pipeline model to generate new rule configuration and validate A/B testing to avoid regression. Hence it is making the scope of steering query optimization for SCOPE users a reality.

Comments:
Why did they choice the contextual bandit and not something like deep reinforcement learning or transformer? Is it because of the production challenges or something else or because its generating one rule hints for each job and heavier models would be overkill? Since they mentioned they wanted to implement multiple rule flip in future, will there be a choice to the customer to select single rule vs multiple rule or will it be enforced?
How reliable is extrapolating the flighting from small number of jobs?
Why would someone use python for implementing the pipeline? Wouldn't a compiled language like C++ work better? Or is it that they are just using python as a wrapper to call C modules.
I know it is an industry paper and they just care for their own datasets. But at least there could have some other readings apart from SCOPE workload, at least one.