Balsa

Query optimization is an under-explored topic in databases. It is complicated process and can be a costly affair. Many relational systems use a heuristic based approach instead of searching the entire search space because of the cost associated to it. Balsa uses a deep reinforcement learning method which is based on the reward model. It also does not require an expert optimizer(unlike Neo) neither expert cost model(DQ) as reference and instead uses a simulator to avoid disasters and during the reality phases, produces high performance plans. To avoid getting stuck into local minima during the best-first beam search, it uses safe exploration approach that picks from set of probably good plans(top k plans). It also has timeouts to avoid getting stuck onto bad plans(unlike the typical case where wrong decision ends the game faster in RL). Also this features boast its generalization and avoids overfilling on data. Hence, Balsa is able to successfully work on unseen queries as well. It uses on-policy learning to make training faster than its counterpart Neo. To enhance generalization, it uses a diversified experience algorithm. Hence Balsa is a fresh approach towards query optimization using reinforcement learning.

Comments:
I really liked the paper layout, the problem statement was clearly defined and the problem faced was also very clear. Really appreciated them giving the GitHub link in the paper itself. Also since they have the simulation to work on real-world data and not just trained on the generic benchmarks like TPC.
Since Balsa is based on deep learning, what is the carbon footprint of it?
Is the power consumption and gpu cost justified for the performance benefits (upto 2.8x)?
Also unlike MPDP that can also work on CPU, Balsa has the architectural dependency of the GPU. Maybe, if they can use something like OpenVino framework to utilize the CPU instead but not sure how effective it can be.
How is the value of k in k-plans decided? Shouldn’t exploring all the k plans one of the reason that might have hindered its performance? Maybe if substituted by a better algorithm here, they might be able to improve their performance even better. Or are they somehow utilizing the gpu to explore all the k plans at once so under a certain threshold, we won’t see much difference?
They mentioned that transformers produced similar results but required high computational cost. But after the advent of newer models like BERT and GPT-3 and even more work pouring in due to the fight between the tech giants like Microsoft and google to improve their search engines, it might be a good idea to explore this space cause in order to scale, they will be working on reducing the computational cost. Also, special accelerators just for transformers and similar infrastructures will probably on its way.
They mentioned that they outperformed a commercial system as well. Are they talking about Microsofts QO-advisor that is based on contextual bandit which is

kind of similar to reinforcement learning?