Fairness in Serving Large Language Models

Summary

With an application like ChatGPT it recieves a natural language query. Then it has some model or program in other words that takes the query as input and spites out a answer. That answer is then sent over the network back to the client. This paper addresses how to make that process fair. Some of the trade-offs are computation power vs fair. For example you could come up with a execution plan that is fair but does not utilize all of the computing power.

Current solutions use a first come first server solution. They impose a request per minute limit so that one client making a lot of request does not hog the server. A solution that is preposed for networking and os scheduling is fair queueing. Fair queueing ensure that each client gets $1/n$ resources. This paper applies fair queueing to llms. It does this at the token level instead of the request level. One of the challeges is that the input size and output size are not know before hand. This is not true for the other applications of fair queueing like networking. They prepose a Virtual Token Counter (VTC) the compare the cost of different requests. They think that they are the first people to address this problem.

Though request can be answer one at a time it is much more efficeint to answer a lot of them together. This is an important quote from the paper: “Intuitively, VTC tracks the services received for each client and will prioritize the ones with the least services received, with a counter lift each time a client is new to the queue.”

Pros

- It uses an existing solution to solve a current problem.
- Uses token to track fairness instead of requests.

Cons

- Does not talk about sever sturcture of LLM applications.
- Had trouble understanding the results. It seems a rather difficult problem to show the improvements.

Further Developments

There are a lot of room for development in this area because there are not any llm specific fairness staratigies.

Other Comments

I have question about the underlying network architeuture of this system. What is it? I think that part of the answer is that there is fairness across multiple machines that are all running the LLM. There is also fairness at each machine which is batch tokens.