Disciplina: INF16179 - Sistemas Distribuídos
Alunos: Thiago da Silva Meireles de Souza / Marcio Merçon de Vargas

Trabalho I

Metodologia

Este trabalho implementa o aprendizado federado por meio da troca de mensagens via publish e subscribe com uso do broker EMQX. O broker usado está rodando localmente com auxílio do docker, ele está configurado para permitir mensagens de até 100mb. Foram criados dois programas, o primeiro, *client*, usado para a resolução das tarefas de aprendizado, o outro foi um *server*, responsável pela distribuição das tarefas e agregação dos resultados. Além disso, foram criadas duas classes, *mqtt*, responsável por toda comunicação e a*prendizado*, responsável por todas as tarefas do aprendizado federado.

Para os testes foram realizados aprendizados com 2, 4, 6, 8, e 10 clients com um máximo de 20 rouns e meta de accuracy de 100%. A execução foi realizada por meio de um script, que automatiza todo o experimento. Ao fim também foi coletado o resultado final considerando a avaliação com todo o dataset usado.

Resultados Obtidos

A figura 1 foi obtida com accuracy apenas com uso dos dados do server, já a figura 2 foi obtida com base na accuracy obtidas por todos os clientes. De maneira geral, foi possível observar que a velocidade de conclusão do aprendizado foi mais rápida que a forma convencional com comunicação via ray e obteve bons resultados. Na figura é possível notar que a qualidade do modelo quando usando poucos dados é menor. Na figura 2, ao aumentar o número de dados conjuntamente com o número de clientes, houve uma variação de quase 2% de 2 para 8 clientes.

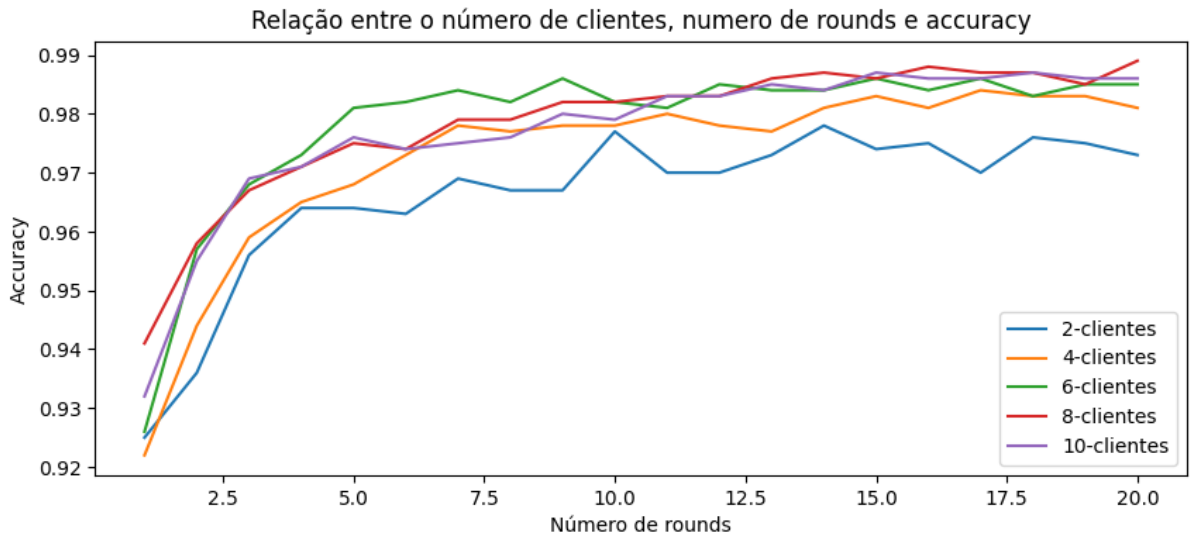

Figura 1

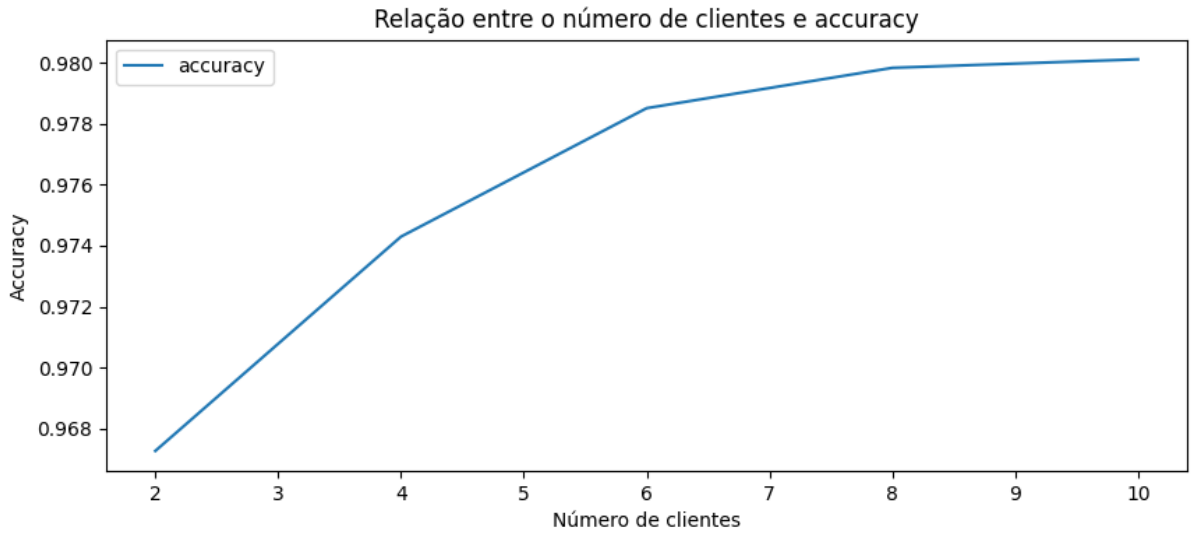

Figura 2

Discussões

A implementação do aprendizado via publish e subscribe, embora possua menor tempo de execução traz alguns problemas na sincronização e tolerância a falhas. Há problemas como a perda de mensagens e a inoperabilidade de um cliente, o que se torna uma tarefa mais complexa de ser detectada. Por outro lado, apesar das dificuldades de sincronização, tem-se a vantagem de não exigir uma conexão direta entre o server e o client.