ANN-RSFK: Busca genérica de similaridade em GPU

Bruno Henrique Meyer[1], **Aurora Pozo**[1], **Wagner M. Nunan Zola**[1]

[1]Departamento de Informática – Universidade Federal do Paraná (UFPR)
Curitiba/PR

Resumo. *Este artigo apresenta o algoritmo ANN-RSFK, que utiliza como base o algoritmo RSFK baseado em árvores para resolver problemas de buscas de similaridades em GPU. A proposta foi comparada com estratégias presentes na biblioteca FAISS, que também utilizam GPU para executar os algoritmos. O ANN-RSFK apresentou melhor custo-benefício entre tempo e qualidade para baixos valores de acurácia, e demonstra potencial para superar as estratégias da biblioteca FAISS em diversos cenários.*

1. Introdução

Algoritmos de aprendizado de máquina possuem aplicações em diversas áreas do conhecimento. Um tipo de problema relacionado às essas aplicações é o *Approximate Nearest Neighbors (ANN) search*, ou busca de vizinhos aproximados. Neste problema, uma base de dados de referência é utilizada para identificar os pontos mais próximos de cada outro ponto pertencente a uma segunda base de dados chamada consulta. Devido à complexidade computacional dos algoritmos que resolvem o problema de *ANN search*, diversos tipos de estratégias podem ser necessárias para acelerar o algoritmo, como o paralelismo em CPU ou em GPU. Diversas técnicas e algoritmos podem ser utilizadas para resolver problemas de ANN *search* [Aumüller et al. 2017]. De acordo com o trabalho citado, a biblioteca FAISS [Johnson et al. 2017] é a melhor escolha para resolver o problema mencionado anteriormente com o paralelismo em GPUs. A biblioteca FAISS tem o código aberto, e emprega a técnica de *inverted files*. Em trabalhos passados, o algoritmo RSFK foi proposto e se tornou parte do estado da arte para solução do problema de construção de KNN *Graphs* em GPU [Meyer et al. 2021b, Meyer et al. 2021a]. No presente trabalho desenvolvemos o algoritmo ANN-RSFK, que resolve problemas genéricos de ANN *search* em GPUs adaptando o algoritmo RSFK. O algoritmo RSFK é utilizado apenas para criar KNN *Graphs*, onde a base de consulta e a referência é necessariamente a mesma, já o ANN-RSFK pode ser utilizado para fazer buscas genéricas.

2. ANN-RSFK

O ANN *search* é um problema onde se busca em conjunto de referência B, os K pontos mais próximos de cada ponto $x \in Q$, onde Q representa o conjunto de pontos de consulta. Esses pontos possuem D dimensões e pode-se considerar diversos tipos de espaços métricos, comumente a distância euclideana. A versão força bruta para resolver esse problema calcula várias distâncias e tem complexidade computacional $\mathcal{O}(|B| \times |Q| \times D)$. Por esse motivo diversas estratégias de aproximação podem ser utilizadas para reduzir o número de distâncias computadas e acelerar o algoritmo. O algoritmo ANN-RSFK visa utilizar as estratégias de aceleração do algoritmo RSFK para criar árvores que dividem os hiperplanos e a base de dados de referência no problema de *ANN search*. Com isso, uma árvore pode ser utilizada para identificar um conjunto de pontos que serão candidatos a vizinhos de cada ponto da base de consulta. O ANN-RSFK utiliza uma estratégia *warp-centric* para computar as distâncias e atualizar as vizinhanças [Meyer et al. 2021a].

3. Métodos

Os experimentos deste trabalho utilizaram um computador com uma GPU NVIDIA RTX 2070, processador Intel i5-4460 e 16 de memória RAM (CPU). O ANN-RSFK foi comparado com quatro estratégias implementadas para usar GPU na biblioteca FAISS (FLATL2, IVFFLATL2, IVFPQ). A estratégia FLATL2 consiste em um algoritmo que computa os vizinhos exatos de cada ponto da base de consulta. Para cada estratégia foi medido o tempo de execução e a acurácia relativa aos vizinhos exatos. A base de dados utilizada foi a GoogleNews300 que possui 3000000 pontos com 300 dimensões, que foram reduzidas para 32 utilizando o algoritmo de redução de dimensionalidade PCA.

4. Resultados e Conclusão

A Figura 1 apresenta os resultados do experimento. Percebe-se que o algoritmo ANN-RSFK possui melhores custo-benefícios entre tempo e acurácia quando comparado à estratégia IVFPQ. Para acurácias menores que 50%, o método ANN-RSFK foi o melhor algoritmo, porém a estratégia IVFFLAT ainda tem o menor tempo de execução para acurácias mais altas. Diversas melhorias algorítmicas e estratégias para uso eficientes dos recursos da GPU podem ser adicionadas no ANN-RSFK. Dessa forma, a estratégia apresentada neste trabalho pode ser considerada promissora e competitiva em relação às outras opções disponíveis no estado da arte para problemas de ANN *search*.

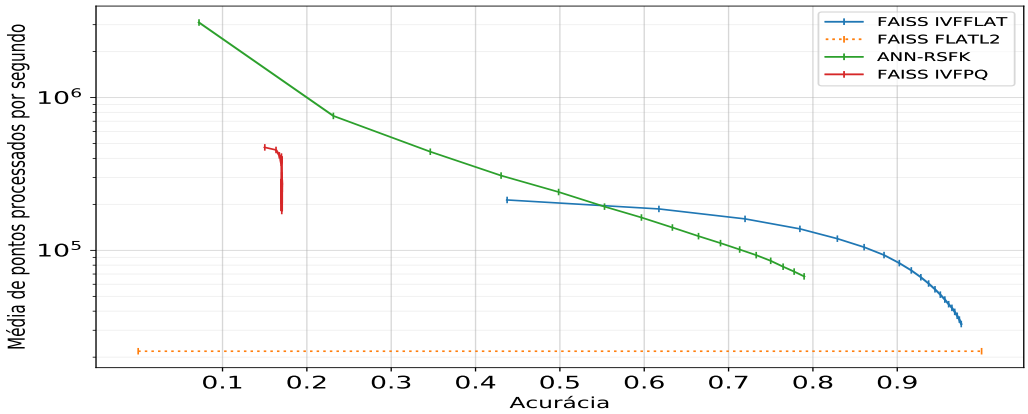

Figura 1. Vazão e acurácia obtida para o uso de diferentes métodos variando seus parâmetros (cada ponto representa uma execução).

Referências

Aumüller, M., Bernhardsson, E., and Faithfull, A. (2017). Ann-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. In *International Conference on Similarity Search and Applications*, pages 34–49. Springer.

Johnson, J., Douze, M., and Jégou, H. (2017). Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Meyer, B., Pozo, A., and Nunan Zola, W. M. (2021a). Warp-centric k-nearest neighbor graphs construction on gpu. In *50th International Conference on Parallel Processing Workshop*, pages 1–10.

Meyer, B. H., Pozo, A. T. R., and Zola, W. M. N. (2021b). Improving barnes-hut t-sne algorithm in modern gpu architectures with random forest knn and simulated wide-warp. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 17(4):1–26.