Bachelor/Master Thesis

Efficient Inference of Random Forests on Modern Memory Architectures

Christian Hakert
Dr.-Ing. Kuan-Hsun Chen
Otto-Hahn Str. 16
Technische Universität Dortmund
Email: christian.hakert@tu-dortmund.de
February 10, 2021

Decision trees are a fairly simple machine learning algorithm, which operate on labeled input data [2]. The input data is interpreted as a multi dimensional data object. At each node of the decision tree, one dimension of the input is compared to a fixed threshold value. Depending on the outcome of the comparison, either the left or the right sub-tree is traversed, which again contains tests on other dimensions. The leafs of the tree contain outputs of the machine learning model, which can be either classification or regression outputs.

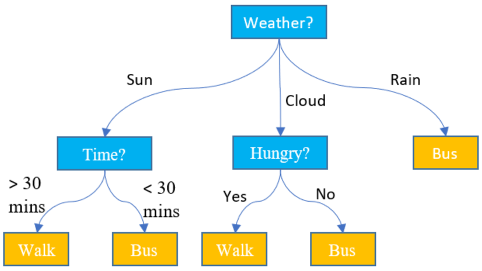

Figure 1: A simple decision tree

However, in the age of big data, a set of decision trees as random forests are often used to analyze massive high dimensional data, which leads to megabyte or even gigabyte sized models. These trees are usually stored in the main memory and processed directly out of the memory. For modern computers, the speed of such a memory intensive application mainly is determined by the use of the various caches of the CPU. For example, the memory layout of decision trees was leveraged to keep the most often accesses paths of the tree in a cache-aware manner [1]. This work in fact brings more attentions on how to deploy tree ensembles for the underlying memory architecture.

When the multicore embedded systems are in use, several technical challenges have to be taken into consideration: cache coherence, memory bandwidth and eventually non-uniform memory access (numa) like effects with hybrid memory architectures. All these effects become embarrassingly important when not only read accesses happen to the model, but also write accesses overlap the read accesses. Such write accesses could be potentially caused by storing important profiling information or even by updating single decision tress during runtime. In all these scenarios, the absolute positions of decision trees in memory, the relative positions and the access order cause important additional memory latencies, which can be reduced by carefully placing decision trees in main memory in an optimized placement.

In this thesis, students first should get familiar with the specifics of the memory subsystem and the implementation of decision trees and profile the runtime bottleneck of tree executions on targeted systems or suitable simulators. Along with the profiling results, students should propose suitable treatments (not restricted in any layer) under analytical arguments to optimize the runtime of decision trees without changing the structure of nodes dependency. Eventually, the proposed approaches should be extensively evaluated and compared with some baselines to reach a reasonable conclusion.

Other suggestions and related topics are also welcome. Please do not hesitate to make an appointment.

Required Skills:

- Knowledge of computer architecture
- Basic knowledge of machine learning
- Comfortable in Python, C and C++ programming

Acquired Skills after the thesis:

- Knowledge about modern computer architectures
- Deep understanding of decision trees
- Experience of research campaigns

[1] Buschjager, S., Chen, K. H., Chen, J. J., & Morik, K. (2018, November). Realization of Random Forest for Real-Time Evaluation through Tree Framing. In 2018 IEEE International Conference on Data Mining (ICDM) (pp. 19-28). IEEE.
[2] Lucchese, C., Nardini, F. M., Orlando, S., Perego, R., Tonellotto, N., & Venturini, R. (2017, September). Quickscorer: Efficient traversal of large ensembles of decision trees. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 383-387). Springer, Cham.