Classifying Performance Bounds Using Machine Learning

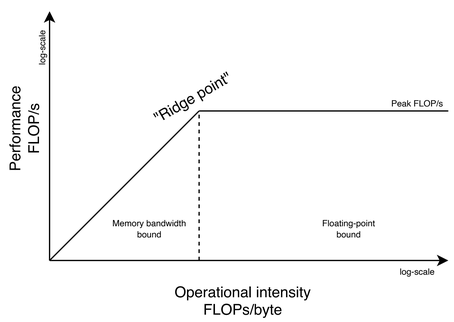
Lewis Littman and Tom Deakin
University of Bristol, Bristol, UK

Abstract

Traditional performance analysis tools, such as the Roofline model, require visual interpretation to determine performance bounds. For CPUs which have complex cache hierarchies and front-end out-of-order capabilities—that is the CPUs we use for high performance computing—accurately identifying the true performance bound is challenging. This work is the first steps towards a data-driven approach to performance modelling, leveraging Machine Learning techniques. We build and evaluate a number of supervised and unsupervised models using a new curated data set of performance counters collected from well-understood (i.e., easily labeled) benchmark applications. We further analyse the data set and highlight potential "performance fingerprints" obtainable using this methodology.

Performance Limiting Factors

The Roofline model (Williams, Waterman and Patterson) categorises the performance limiting factor of an application/kernel by considering how its performance (in FLOPS) compares to theoretical peak on that processor. In particular, based on the Arithmetic Intensity—the ratio of floating-point operations to data movement—that kernel will be classed as "compute bound" or "main memory bandwidth bound" based on which side of the *ridge point* it falls. In order for the categorisation to be considered correct the performance should be close to the roofline bound, often cited as 50%.



Access the data set

`https://doi.org/10.5281/zenodo.17194638`

Data Set generation

We collected a data set of performance counters by executing each of the following applications/problem sizes. Each application was run 3 times to collect all performance counters, and together they form a single record in the data set. 100 records for each application/problem size were collected, resulting in 1,200 records in the new data set. Performance bounds were verified by Roofline analysis even though they are well known.

Benchmark	Problem Sizes	Performance Bound
SGEMM	15k-by-15k and 10k-by-10k	Compute
DGEMM	15k-by-15k and 10k-by-10k	Compute
miniBUDE	128 PPWI, WGS 1	Compute
STREAM	10M elements	Main Memory Bandwidth
LBM D2Q9	256-by-256 and 1024-by-1024	LLC Memory Bandwidth
MiniFE	50-by-50-by-50	Main Memory Bandwidth
3D Heat stencil	120-by-120-by-120	LLC memory bandwidth
LU Solver (MKL)	8192-by-8192 and 16384-by-16384	LLC memory bandwidth

Performance counters were collected using Linux `perf` and normalised. Each record in the data set therefore contains:

- GFLOPs
- FLOPc
- IPC
- retiring %
- bad speculation %
- frontend bound %
- backend bound %
- vectorised SP instruction ratio
- vectorised DP instruction ratio
- cache miss ratio
- L1 cache miss ratio
- L2 cache miss ratio and L3 cache miss ratio

Baseline Models

Three naive models were tested to create a baseline for the machine learned models. The Uniform model randomly predicts the performance limiting factor as "compute bound" or "bandwidth bound" with equal probability. The Proportional model predicts these limits based on their relative proportion in the dataset. The Majority model will always predict "bandwidth bound", as the most frequent in the dataset.

Class	Uniform			Proportional			Majority		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Bandwidth Bound	0.56	0.42	0.48	0.64	0.59	0.62	0.58	1.00	0.74
Compute Bound	0.40	0.54	0.46	0.49	0.54	0.51	0.00	0.00	0.00
Accuracy		0.47			0.57			0.58	

As can be seen, these baseline models are essentially no better than random.

Machine Learning Models

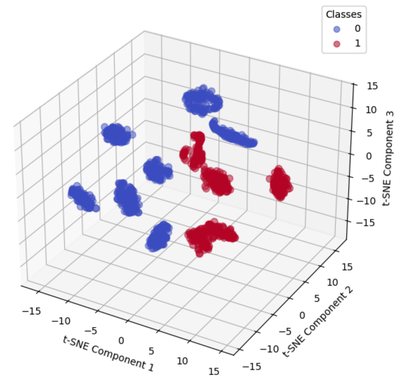
We train a number of models using this data set. With default parameters, they result in perfect accuracy (explained by the separability of the data set as shown in the t-SNE analysis; see right). We mitigate this overfitting with leave-one-out cross-validation. The Multilayer Perceptron Model (MLP) had one hidden layer of 50 neurons with the ReLU activation function and trained with the Adam optimiser.

Model	Average Accuracy
Decision Tree	0.833
k-NN	0.917
Logistic Regression	0.917
Random Forest	0.833
SVM	0.917
MLP	0.917

These models show high levels of accuracy (91%), a significant improvement of the baseline (random) model. We believe this method can be extended with additional classes, such cache bandwidth bound and memory latency bound.

t-Distributed Stochastic Neighbour Embedding analysis

The t-Distributed Stochastic Neighbour Embedding (t-SNE) analysis reduces the dimensionality of the dataset to 3D while preserving (linear and non-linear) relationships.



Memory bound (**Class 0**) and compute bound (**Class 1**) are clearly separable, explaining the machine learning model classifications.
The clusters represent an application and problem size configuration. They are distinct, indicating that applications possess an intrinsic **"performance fingerprint"** capturing their unique performance characteristics. We are looking to further the research into this in future work.