Predicting Personal Computer Performance Based On Component Specifications

Zarin Loosli

If this project is selected I would like to be a part of it

1. Problem Description

Custom-built Personal Computers (PCs) pose a somewhat unique challenge in the world of computing performance. While most computing devices that people interact with on a daily basis (phones, laptops, game consoles, etc.) use a specific set of components that have been validated to work well together, PCs can come in a nearly-infinite number of possible configurations. Each PC contains several components: CPU, CPU cooler, motherboard, RAM, storage, and power supply are all standard, as well as a GPU that may be a separate processor or may be integrated with the CPU. Each of these components can vary immensely in terms of quality and performance. What's more, the separation of concerns in a modern PC can lead to performance bottlenecks, where top-of-the-line hardware is held back by a single component that cannot keep up with the rest. This project would seek to create a machine learning algorithm that can predict the performance of a system based purely on static analysis of its constituent parts. This would allow consumers to confidently make purchase decisions, including for systems costing thousands or tens of thousands of dollars, while having confidence that they are making a wise purchase decision.

2. Potential Features

A huge number of features could be fed into the model. This includes GPU specifications, CPU specifications, and information about other parts of the system like available storage and memory. The performance would be returned as an estimated score in a predetermined benchmark. As an example data set:

GPU Memory (MB)	24,576
GPU Memory Type	6.5[1]
GPU Memory Bus (bits)	384
GPU Memory Bandwidth (GB/s)	1008
GPU Clock frequency (MHz)	2,235
GPU Memory clock frequency (MHz)	1,313
GPU TDP (W)	450
Shading Units	16384
TMUs	512
ROPs	176
SM Count	128
Tensor Cores	512
RT Cores	128

[1] GPU memory types are in the form `GDDR[1-9]+X?` To sanitize this for our algorithm, we would strip away the extraneous letters and use a 0.5 to represent the presence or absence of an X. So "GDDR4" would become "4" and "GDDR8X" would become "8.5"

L1 Cache (KB)	128
L2 Cache (MB)	72
CPU Clock frequency (MHz)	3,000
Physical processors	1
Logical processors	32
# of cores	24
Manufacturing process (nm)	7
CPU TDP (W)	125
System Memory (MB)	16,384
Memory Type	4
Memory Frequency	3,000
Hard drive capacity (GB)	1,000
Performance Score	**29 837**

3. Sources And Means Of Collection For Training Data

The 3DMark Time Spy benchmark provides both a real-valued quantification of overall system performance and a large database of prior test results. As far as I am aware, this data is not packaged to be downloaded in its totality, but it could be scraped using python libraries or NodeJS and Puppeteer. It could then be supplemented with details from TechPowerUp when 3DMark does not report specific attributes of the hardware being used. In the example above, TechPowerUp was used to find the rated TDP and core counts for the GPU.