White-Box OLAP Performance Modeling for the Cloud

Maximilian Kuschewski
maximilian.kuschewski@uni-jena.de
Uni Augsburg, LMU, TUM

Viktor Leis
viktor.leis@uni-jena.de
Friedrich-Schiller-Universität Jena

ABSTRACT

Public cloud providers offer hundreds of heterogeneous hardware instances. For cloud-based analytical query processing systems, this presents a major challenge: depending on the hardware configuration chosen, performance and cost may differ by orders of magnitude. We explore this relationship by building a white-box model that takes the workload, hardware, and cost into account to determine the optimal instance configuration. We argue that such an approach can guide the evolution of cloud-native OLAP systems.

1 INTRODUCTION

Public cloud providers offer a large number of hardware instances that can be rented, started, and stopped on demand within seconds. Consequently, selecting the hardware that database systems run on represents an additional degree of freedom when building systems, adding features, and planning queries. We need a way to explore this new degree of freedom and how it can be used to optimize query execution. However, we believe black-box models are unsuited for this task because they provide little insight and model specific systems. Instead, we propose using white-box models that explicitly describe the relationship between workloads and the cloud instances they are executed on.

We have designed such a white-box model with a focus on OLAP workloads. An interactive tool for exploring the model predictions is available online [1, 2]. Given some abstract description of an OLAP workload, our model predicts the monetary workload cost for the execution on available cloud instances. This cost also serves as the optimization objective, as execution time may be (almost) arbitrarliy small due to on-demand hardware scaling. Our model includes multi-layer instance-local caching and materialization, as well as the imperfect scaling and network data exchange required for describing distributed processing.

2 USE CASES

There are multiple use cases for our model.

Cost Optimality As A Benchmark For New Systems. We do not model performance bottlenecks and weaknesses of existing database systems, and assume that all available hardware resources can be fully exploited. Of course, this means that building a new cloud-native query processing engine that performs as well as our model is challenging. However, we only rely on well-understood database concepts like caching, distributed query processing, and hybrid hash join-like operators. Thus, the hurdles are mainly a matter of engineering and systems building. We believe that, during system design and development, our cost optimality model is a useful benchmark against which a new system can be compared.

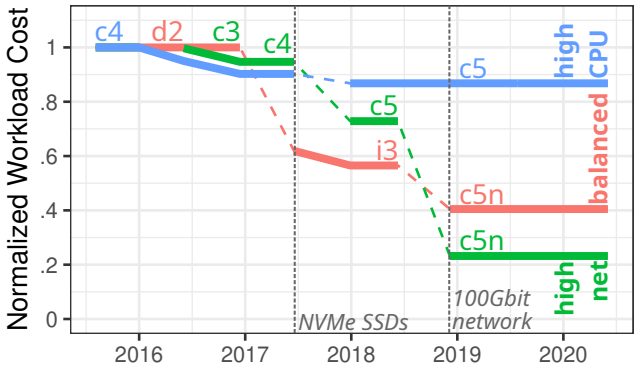

Figure 1: Cheapest cloud instance type for three different workloads over 5 years as predicted by our model

Evidence-Based Performance Feature Prioritization. By disabling features like multi-layer caching or distributed processing in the model, system architects can quantify how large the performance and cost benefit of these features is. This enables evidence-based decisions about prioritizing performance optimizations.

Hardware/Software Co-Evolution. The available hardware in the cloud is not static but changes over time. Our model allows reacting to new hardware opportunities by recomputing the model with the new hardware data. We can illustrate this idea by applying our model to hardware configurations and prices from the past 5 years. Figure 1 retroactively shows how the costs for three different workloads (CPU-bound, balanced, and network-bound) according to the model. Since 2015, there have been two major relevant changes in the EC2 hardware landscape. The first was the introduction of fast NVMe SSDs in mid 2017. This had a significant effect on the balanced workload, which moved from an instance with 20 disks (d2) to an instance with 8 NVMe SSDs (i3) – almost halving workload cost. The second major change was the introduction of 100 Gbit network instances at the end of 2018, which had even larger consequences: the cost of the network-bound workload dropped to a quarter of the initial cost (and even the balanced workload switched away from i3 to c5n). The CPU-bound workload, in contrast, did not see large gains in our 5-year period – which we find surprising given the rising number of CPU cores in commodity servers. This historical example illustrates how our model can be used to react to changes in the hardware landscape by quantifying the savings from using new hardware. This can then be compared with the engineering cost required to exploit the novel hardware.

REFERENCES

[1] 2020. https://github.com/maxi-k/costoptimal-model.
[2] 2020. https://maxi-k.shinyapps.io/costoptimal/.

CIDR '21, January 10–13, 2021, Chaminade, USA