Operating Systems Design: Critique

2019712600
Oh, Seungmin

mClock: Handling Throughput Variability for Hypervisor IO Scheduling

VMs are currently used for a variety of computing and research purposes. As a result, various environments can be controlled, but the control of IO-related resources is still very limited. Efficiently controlling IO by the VM hypervisor is an important and difficult problem. This paper proposes a new algorithm *mClock* to solve this problem, *dmClock* for distributed storage environment, and also proves the performance experimentally.

The main advantages and the main points of the paper are that they studied resource regulation of hypervisor which was not studied well, and considered distributed storage. The limitation of this paper is that the shape of the recent VM environment is changing. In addition to the recent VMs, containers and Unikerenl should be considered.

In future papers, I would like it to be mClock on IO for container environments. This is because the recent increase in the container environment and the large cluster environment using the same may produce another result.

FlashFQ: A Fair Queueing I/O Scheduler for Flash-Based SSDs

The SSD environment has different characteristics from that of the existing HDD storage. Existing simple time division schemes become less reactive. To solve this problem, the paper proposes Flash, an SSD-based scheduling method. This approach yields as much parallelism and performance as possible, but with very few violations of fairness.

The advantage of this paper is that it fundamentally looks at the difference between HDD and SSD environments and based on this, it considers both main points (parallel and fairness). On the other hand, the limit is that the performance is still worse than other CFQ and SFQ in some cases.

In future papers, it is worthwhile to analyze the differences in the NVMe environment. And as various flash-based storages are coming out, I hope to develop techniques in other researched and commercialized storage environments.