Automated I/O Parameter Tuning of Scientific Applications with Parametrizable Workload Replays

Azat Nurgaliev, Marcus Paradies
German Aerospace Center, Institute of Data Science

1 Introduction

NVMe SSDs offer unprecedented throughput and latency performance for various data-intensive, scientific application domains such as earth observation, radio astronomy, and weather forecasting. Unfortunately, even though data analysis pipelines from these domains often operate on large data sets on the order of several TBs, most of them fail to take advantage of the parallelism and bandwidth offered by NVMe SSDs due to their single-threaded I/O implementations. Moreover, when scientists do parallelize their pipelines, they typically focus on optimizing the computation instead of IO. To make things worse, the growing heterogeneity of available storage devices with respect to their latency and throughput characteristics makes I/O parameter tuning (e.g. degree of parallelism, block size) a challenging task for non-experts [3,5,6,9].

As such, we see an opportunity for improvement. In this work, we propose an automated I/O parameter tuning system for black-box scientific applications, which finds and reports the optimal I/O parameter configuration for a given application and storage device combination.

Normally, I/O parameter tuning requires access to the source code of the data analysis pipeline to modify the I/O parameters. In cluster infrastructures, however, containerization makes such a white-box approach infeasible since the source code is not available. To overcome this problem, we propose a different I/O parameter tuning approach that operates in two phases: (1) capturing the I/O traces of an application, and (2) replaying the traces with different I/O parameters to identify the optimal configuration.

Our approach differs from the existing I/O trace replay tools (e.g., `blkreplay`, `btreplay`, `hfplayer`), as they focus on replaying in different system environments [2,4,7] while our approach is about replaying with different parameters. To achieve that, we resort to synthetic I/O workload generators (e.g. FIO, FILEBENCH), through which I/O operations can be easily simulated with varying parameters on the underlying storage devices [1,8].

2 Proposed idea

We use fine-grained I/O tracing, generate a parametrizable I/O workload replay using a synthetic I/O workload generator, and run multiple parametrized workload replays on the target storage hardware to determine the optimal I/O parameter combination. In the context of this work, we focus on two fundamental application I/O parameters, namely the level of I/O parallelism and the data block size. For an optimal parameter tuning, we consider a balance between both parameters, i.e., larger block sizes could result in a lower degree of parallelism while smaller block sizes demand a larger degree of parallelism. In contrast to other I/O workload replay tools, which focus on replaying workloads on different hardware setups, our focus is on replaying the workload always on the same storage setup, but with different application I/O parameter configurations. Our approach consists of three phases: (1) I/O trace collection, (2) I/O workload replay creation, and (3) application I/O parameter tuning using parametrizable I/O workload replays. To trace the I/O workload we use `blktrace`, which collects detailed information about every single I/O operation and also captures times spent on computations. Specifically, we collect the following I/O characteristics: the size of reads & writes, distribution between random and sequential operations, distribution of request size, and delays between I/O requests. In order to create a parametrizable I/O workload replay, we use `fio` and create an (oftentimes complex) `fio` job file, which precisely mimics the I/O workload of the application. The advantage of `fio` is that through the generic job description, application parameters can be directly adjusted in the job file, which allows rerunning the same I/O workload under different I/O parameter configurations.

Summary and Future Work: We propose parametrizable I/O workload replays, which enable optimal I/O parameter tuning of black-box scientific applications running on modern NVMe SSDs. For future work we plan to improve the accuracy of the I/O workload replay for complex, parallelized I/O workloads with interleaved computations.

References

[1] AXBOE, J. Fio-flexible io tester. *URL http://freecode. com/projects/fio* (2014).

[2] BRUNELLE, A. D. btrecord and btreplay user guide, 2011.

[3] CHEN, F., LEE, R., AND ZHANG, X. Essential roles of exploiting internal parallelism of flash memory based solid state drives in high-speed data processing. In *2011 IEEE 17th International Symposium on High Performance Computer Architecture*, pp. 266–277. ISSN: 2378-203X.

[4] HAGHDOOST, A., HE, W., FREDIN, J., AND DU, D. H. C. On the accuracy and scalability of intensive i/o workload replay. pp. 315–328.

[5] HE, J., KANNAN, S., ARPACI-DUSSEAU, A. C., AND ARPACI-DUSSEAU, R. H. The unwritten contract of solid state drives. In *Proceedings of the Twelfth European Conference on Computer Systems*, EuroSys '17, Association for Computing Machinery, pp. 127–144.

[6] KAKARAPARTHY, A., PATEL, J. M., PARK, K., AND KROTH, B. P. Optimizing databases by learning hidden parameters of solid state drives. 519–532.

[7] SCHBEL-THEUER, T. blkreplay and sonar diagrams, 2012.

[8] TARASOV, V., ZADOK, E., AND SHEPLER, S. Filebench: A flexible framework for file system benchmarking. *USENIX; login 41*, 1 (2016), 6–12.

[9] WU, K., ARPACI-DUSSEAU, A., AND ARPACI-DUSSEAU, R. Towards an unwritten contract of intel optane SSD. 8.