RezSort: Sorting 1TB using Energy-efficient NVMe SSDs

Waleed Reda
Université catholique de Louvain
KTH Royal Institute of Technology

Dejan Kostić
KTH Royal Institute of Technology

1. Introduction

This paper summarizes the configuration, benchmark steps, and results of a system that was built to improve upon the 1TB JouleSort benchmark [8] (Daytona Category). We use a desktop machine equipped with energy-efficient SSDs and show that we can improve upon the current 2013 [4] and 2019 [6] JouleSort world records by 17% and 15%, respectively.

Our desktop machine is a modified Asus ROG Strix G15CK build [2]— equipped with Intel Core i7-10700 CPU running @ 2.9 GHz, and paired with 16 GB DDR4 DRAM. We use 2× 1TB SK hynix Gold P31 NVMe SSDs and a 2TB Samsung 980 Pro NVMe SSD. Using this hardware, we are able to perform a 1TB sort (with 10^{10} records) in an average time of 1379 seconds running at 100.4 W—requiring a total of 138,416 joules.

2. Hardware

The last few years have witnessed a large improvement in I/O performance of storage media. NVMe SSDs in particular have been gaining popularity as they use high-bandwidth PCIe interfaces and provide low IO latencies. Since IO is often a bottleneck for 1TB sorts (since the dataset does not fit into memory), high-performance storage media can help boost their performance. Secondly, SSDs are becoming more and more energy-efficient [5] —providing another avenue for improving JouleSort performance. We explore how big that gap has gotten with next-generation NVMe SSDs. For our build, we specifically picked SSDs that scored highly in energy-efficiency with different types of IO benchmarks [9, 10].

System specs. Our system is based on modified build of an Asus G15CK desktop. This machine is equipped with a power-efficient Intel i7-10700 CPU ("Comet Lake") paired with 16 GB of DDR4-2933 DRAM. The processor has 8 cores (16 with hyper-threading) running @ 2.9 GHz and a TDP of 65 W. The motherboard (Asus ROG) uses the Intel B460 Chipset and houses 2× M.2 SSD slots, 2× PCIe 3.0 ×16 slots, 1× PCIe 3.0 ×1 slot, and 6× SATA 6.0 Gb/s ports. For storage, we use 2× 1TB SK hynix Gold P31 NVMe SSDs and a 2TB Samsung 980 Pro NVMe SSD. Given that our motherboard is limited to only 2× M.2 slots, we use a PCIe 3.0 ×4 to M.2 adapter to house the third SSD. We did not experience any notable performance degradation in our sort runs as a result of using an adapter.

The system is equipped with a 500W Great Wall Power Supply and we use all the stock cooling options provided by the G15CK desktop. To conserve energy, we disable the RGB LEDs on the Asus motherboard. The system has an idle power of roughly 23 W.

Pricing. All hardware components used in our system are (or were) commercially available. The prices of these components are listed in Table 1.

Part	#	Unit Price	Total Price
Asus G15CK Desktop	1	$1,578	$1,578
SK hynix Gold P31 1TB SSD	2	$195	$390
Samsung 980 Pro 2TB SSD	1	$494	$494
Delock PCIe 3.0 to M.2 adapter	1	$34	$34
smart-me Schuko - Energy meter	1	$122	$122
System total			**$2,618**

Table 1. Component prices (converted from SEK to $).

3. Software

Our system runs Ubuntu 18.04.5 LTS with kernel version 4.15.0-159-generic. Given that external sorts are generally IO-bound, we set the Intel pstate governer to *powersave* mode—which runs the CPU at the minimum frequency.

Storage. Our storage layout is summarized in Fig. 1. We use the two 1TB SK Hynix SSDs primarily for the input/output files, and the 2TB Samsung SSD for storing temporary files generated by the sort. Given that 1TB SSDs cannot host the 1TB sort files on their own (due to file system overheads), we set-up linear logical volumes using LVM which combine each 1TB SSD with 50GB from the 2TB SSD. All volumes use the ext4 file system.

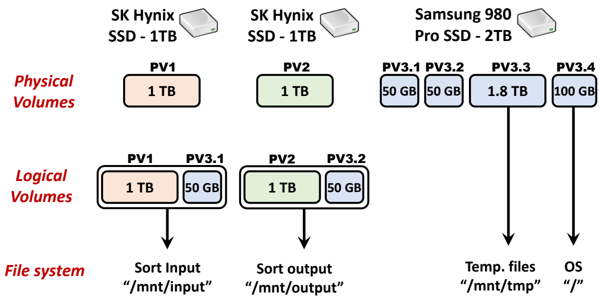

Figure 1. Storage and volume configuration. All volumes are configured to use ext4.

```
nsort -processes=24
-memory=15000M
-method=radix
-format=size:100
-field=name:key,size:10,off:0,character
-key=key
-statistics
-in_file=/mnt/input/unsorted_1tb,direct,
        transfer_size=16M
-out_file=/mnt/output/sorted_1tb,direct,
        transfer_size=128M
-temp=/mnt/tmp,direct,
        transfer_size=32M
```

Figure 2. *nsort* parameters for our best run.

Sort. We use the provided *gensort* tool to create a 1TB input file (10^{10} records). The sort itself is performed using a trial version of *nsort* [1]. We report the parameters for our best run in Fig. 2—which aim to provide a good balance of IO and compute performance. Notably, we set the number of processes such that it is 50% higher than the number of hyper-threaded cores (16 in our case), similarly to NTOSort [4]. This oversubscribed setting can aid in filling the memory controller pipeline as threads may be waiting on memory reads/writes.

4. Measurements

We measure energy consumption using a *smart-me* energy meter [3]. Per the manual, this meter provides an accuracy of ±1%, which satisfies the conditions of the JouleSort competition. The energy meter readings are logged at 1-second intervals along with their associated UNIX timestamps and are automatically sent to a cloud server. We run a simple power collector utility on the same machine as nsort, and it periodically retrieves power readings (at 1-second intervals) recorded throughout the duration of the experiment. The collector uses a simple HTTP GET request to fetch the meter reading corresponding to a given UNIX timestamp. The collector itself is light-weight and has negligible power usage. Similar to prior work [4, 6, 7], we exclude the first and last measurement points for potential fractional readings. After the execution finishes, we terminate the power collector.

We use the collected data to compute the average power consumed during the experiment. We multiply this by the execution time to obtain the total energy used.

5. Results

Table 2 shows the results of five different runs of the sort benchmark, along with their mean and standard deviation. nsort also reports separate statistics for the input and output phases. The input phase takes 52% of the total sort time, uses 1416 MB/s of IO bandwidth, and consumes 645% of CPU. The output phase takes 48% of the total time, uses 1549 MB/s of IO bandwidth, and consumes 442% of CPU. On average, the sort takes a total time of 1379 seconds and uses 138,416 Joules. This equates to 72,249 records sorted per Joule. The average power factor (PF) of our setup for a 1 TB sort run is 0.82. To adhere to the rules of the competition, we also report sort results for a skewed input dataset and show that its performance is comparable to sorting a regular dataset. The Sort Benchmark committee verified the checksums and duplicate record counts for both the regular and skewed datasets.

Compared to the winning entries of the Daytona category for 2013 and 2019 (which are currently in a 2-way tie), our system reduces energy usage by 29,826 Joules (17% better) and 24,739 Joules (15% better), respectively.

	Time(s)	**Power(W)**	**Energy(J)**	**SRec/J**
Run 1	1360	100.6	136,816	73,091
Run 2	1383	100.5	138,992	71,947
Run 3	1382	100.1	138,338	72,287
Run 4	1386	100.5	139,293	71,791
Run 5	1385	100.1	138,639	72,130
avg	**1379**	**100.4**	**138,416**	**72,249**
stdev	9.7	0.2	862.2	452.9
skewed	1366	99.2	135,507	73797

Table 2. Results for all sort runs.

References

[1] Nsort. http://www.ordinal.com/.

[2] ROG Strix GT15 G15CK. https://rog.asus.com/desktops/mid-tower/rog-strix-gt15-series/spec.

[3] smart-me plug - Energy meter. https://www.distrelec.de/Web/Downloads/_t/ds/smart-me_plug_v2_eng_tds.pdf.

[4] A. Ebert. NTOSort. *sortbenchmark. org*, 2013.

[5] B. Harris and N. Altiparmak. Ultra-Low Latency SSDs' Impact on Overall Energy Efficiency. In *12th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 20)*, 2020.

[6] M. Liu, K. Zhang, S. Peter, and A. Krishnamurthy. TaichiSort: Energy-efficient Sorting of 1TB with NVMe and Coffee Lake.

[7] P. Pillai, M. Kaminsky, M. A. Kozuch, and D. G. Andersen. FAWNSort: Energy-efficient Sorting of 10GB, 100GB, and 1TB. *Intel Labs, Carnegie Mellon University*, 2012.

[8] S. Rivoire, M. A. Shah, P. Ranganathan, and C. Kozyrakis. Joulesort: a balanced energy-efficiency benchmark. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 365–376, 2007.

[9] B. Tallis. The Best NVMe SSD for Laptops and Notebooks: SK hynix Gold P31 1TB SSD Reviewed, 08 2020. https://www.anandtech.com/show/16012/the-sk-hynix-gold-p31-ssd-review.

[10] B. Tallis. The Samsung 980 PRO PCIe 4.0 SSD Review: A Spirit of Hope, 09 2020. https://www.anandtech.com/show/16087/the-samsung-980-pro-pcie-4-ssd-review.