TaichiSort: Energy-efficient Sorting of 1TB with NVMe and Coffee Lake

Ming Liu*[1], Kaiyuan Zhang[1], Simon Peter[2], and Arvind Krishnamurthy[1]

[1]University of Washington [2]The University of Texas at Austin

1 Introduction

This paper summarizes our submission for the 2019 1TB JouleSort competition (Daytona and Indy category). We use a moderately high-power desktop system and improve sorted records per Joule by 3.1%, compared to the latest winning 2013 entry.

Our system features an Intel i7-9700 processor, 32GB DDR4 RAM, 3 Intel DC P3600 PCIe NVMe SSDs, and an extra Samsung SATA SSD boot drive. It sorts the 1TB (10^{10} records) dataset in 1432 seconds ($\pm$8.1s) with an average power of 113.9W ($\pm$0.6W). It requires 163,154 Joules ($\pm$247J), achieving 61,292 ($\pm$93) sorted records per Joule. Compared to the winning 2013 entry, this is 5,088 Joules less, an improvement of 3.1%.

2 Hardware

Recent years have witnessed the emergence of high-bandwidth, low-latency storage devices, such as PCIe SSDs. This enables fast data delivery from storage media to the computing unit, holding potential to improve the system energy efficiency when performing external sort. Thus, we configure the following desktop system.

TaichiDesktop. Our system (shown in Figure 1) uses a recent power-efficient Intel i7-9700 CPU ("Coffee Lake"), paired with 32GB DDR4-2666 DRAM. The processor encloses 8 cores (with no hyperthreading) running at 3.0GHz (65W TDP), 32KB L1 I-Cache, 32KB L1 D-Cache, 256KB L2 cache, and 12MB L3 cache. We enable Turbo Boost (maximum clock frequency 4.7GHz) and apply the *Intel_pstate* governor. The mainboard, ASRock Taichi [1], uses the Intel Z390 chipset, containing 3$\times$ PCIe 3.0 x16, 2$\times$ PCIe 3.0 x1, 8$\times$ SATA3, and 3$\times$ Ultra M.2 slots. This provides us enough I/O bandwidth.

For storage, we use 3$\times$ Intel DC 3600 series PCIe NVMe SSD. Two of them are 1.2TB, while the other one is 2.0TB. Each drive is PCIe 3.0 x4. An additional boot drive (Samsung Ultra M.2 SSD, SM951 series, 128GB) is attached to the motherboard.

The power supply is EVGA 450 BT [2], 80+ Bronze rated 450W. We use the default Intel LGA 115x CPU heatsink and fan that comes with the processor.

*Corresponding author. Email: mgliu@cs.washington.edu

Figure 1: TaichiDesktop system.

Part	#	Unit Price
Intel Core i7-9700 Desktop Processor	1	$330
ASRock Z390 Taichi Motherboard	1	$210
Corsair 16GB DDR4-2666	2	$74
Intel DC P3600 PCIe NVMe 3.0 1.2TB SSD	2	$750
Intel DC P3600 PCIe NVMe 3.0 2.0TB SSD	1	$1,920
Samsung SM951 128GB SSD	1	$133
EVGA 450 BT, 80+ Bronze 450W	1	$56
NZXT H710i	1	$200
Total		**$4,497**

Table 1: Price list for TaichiDesktop

System price and power We build the system with commercially available hardware components. Table 1 shows the current retail price (primarily from Amazon.com and Newegg.com). The desktop has 47W idle power and around 163W peak power. The TDP of our Processor is 65W, including the Intel UHD Graphics 630. We turn on the rear fan on the H710i case.

3 Software

Our system runs Ubuntu 16.04.3 LTS with kernel version 4.10.0-28-generic. The system uses a stock configuration in the BIOS and requires no customized drivers. The three Intel PCIe SSDs use the ext4 file system with no specialized configurations. We use the two 1.2TB SSDs for input/output files, and the 2.0TB SSD for temporary results.

```
nsort -processes=4
-memory=24000M
-method=radix
-format=size:100
-field=name:key,size:10,off:0,character
-key=key
-statistics
-in_file=/input/1000g_input,direct,
    transfer_size=16M
-out_file=/output/1000g_output,direct,
    transfer_size=128M
-temp=/sort_temp,direct,transfer_size=64M
```

Figure 2: NSort parameters for the best 1TB sort

We use the provided *gensort* utility to create input data files (with -a option) and validate the results with *valsort*. We use a trial version of *nsort* [6] software for the actual sorting task. Figure 2 reports the Nsort parameters for our best run. Note that we carefully tune the number of process, memory size, input/output/temporary transfer file size in order to balance between the compute capability and storage I/O. Since NSort is a general sort software package, we meet the 2019 designation for the Daytona category, just like previous entries that competed for JouleSort [4, 7, 8].

4 Measurement

We measure the energy consumption during the sort execution using a *Watts Up Pro* power meter [5]. According to its manual, it reads to a precision of 0.1 W and has a specified accuracy of $\pm(1.5\% + 0.3)$W. We connect the power meter to an onboard USB interface and use a public available Linux software [3] to read the power. The utility reads from */dev/ttyUSB0* once per second and we log the data into a file for analysis. The power logger runs on a separate machine (not the machine that executes the sort task). During each test, we first start our power logging software, wait for a few seconds (until some power data has been written to the file), and then run the sort application. After the execution finishes, we terminate the power logger. Similar as the previous work [6,7], we write sort start/end messages into the power log file, and exclude first/last measurement points (for potential fractional reading).

We report the execution time using */usr/bin/time*. We calculate the average power using the log files and then multiply it by the execution time to obtain the total number of Joules.

5 Results

Table 2 presents the results over five runs (including average and standard deviation). Nsort reports the input and output statistics separately. On average across five runs, the input phase takes 625.2s, consumes 348% CPU, and achieves 1613.4MB/s of I/O throughput. The output phase takes 807.1s, consumes 210% CPU, and achieves 1244.6MB/s of I/O throughput. We also observed that the system peak power is 143.7±1.6W.

	Time(s)	Power(W)	Energy(J)	Srec/J
Run 1	1445.3	113.2	163,547	61,145
Run 2	1431.3	113.9	163,005	61,348
Run 3	1423.4	114.7	163,246	61,257
Run 4	1428.8	114.1	162,956	61,366
Run 5	1433.1	113.8	163,015	61,344
Avg	1432.4	113.9	163,154	61,292
stdev	8.1	0.6	247	92.6

Table 2: 1TB sort on TaichiDesktop

Compared to the winning entry from 2013, our results reduce 5,088 Joules of 1TB JouleSort and improves sorted records per Joule by 3.1% for both Daytona and Indy categories.

References

[1] ASRock Motherboard (Z390 Taichi). https://www.amazon.com/ASRock-Z390-TAICHI-Motherboard-Taichi/dp/B07HYP716L/ref=sr_1_2?keywords=asrock+z390+taichi&qid=1566327645&s=gateway&sr=8-2, 2019.

[2] EVGA 450 BT, 80+ Bronze. https://www.newegg.com/p/N82E16817438130?item=9SIA0ZX6CU1988&source=region&nm_mc=knc-googlemkp-pc&cm_mmc=knc-googlemkp-pc-_-pla-beachaudio-_-power+supplies-_-9SIA0ZX6CU1988&gclsrc=aw.ds&&gclid=EAIaIQobChMIuZzS6Jup5AIV0hx9Ch2znAyqEAQYASABEgJJtPD_BwE, 2019.

[3] Watts Up Pro Power Logger. https://github.com/pyrovski/watts-up, 2019.

[4] Andreas Ebert. Ntosort. Technical report, 2013.

[5] Electronic Educational Devices. Watts up? PRO. http://www.idlboise.com/sites/default/files/WattsUp_Pro_ES.pdf, 2019.

[6] Ordinal. Nsort application. http://www.ordinal.com, 2019.

[7] Padmanabhan Pillai, Michael Kaminsky, Michael A Kozuch, and David G Andersen. Fawnsort: Energy-efficient sorting of 10gb, 100gb, and 1tb. *Technical report*, 2012.

[8] Suzanne Rivoire, Mehul A Shah, Parthasarathy Ranganathan, and Christos Kozyrakis. JouleSort: A Balanced Energy-Efficiency Benchmark. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 2007.