FAWNSort: Energy-efficient Sorting of 10GB, 100GB, and 1TB

Padmanabhan Pillai, Michael Kaminsky, Michael A. Kozuch, David G. Andersen
Intel Labs, Carnegie Mellon University

1 Introduction

In this document, we describe our submissions for the 2012 JouleSort competition (10GB, 100GB, and 1TB categories). We have taken a two-pronged approach, experimenting with a low-power, modest speed system and a very fast, moderately high-power desktop system. We have found that we can configure systems that are competitive in terms of energy consumption from both ends of this spectrum.

Our entry for the 10GB (10^8 records) JouleSort category focuses on high performance. It features an Intel® Core™ i7 processor ("Sandy Bridge"), 16GB RAM, two hardware RAID cards, 16 SSDs, and an extra SSD boot drive. It sorts the 10GB dataset in a single pass in just 8.47 seconds (±0.03s) with an average power of 164.4W (±3.6W). It requires 1393 Joules (±32J), achieving 71789 (±1659) sorted records per Joule. This reduces energy and improves sorted records per Joule by 2.6% compared to the winning 2011 10GB Daytona/Indy entry.

For the 100GB (10^9 records) JouleSort competition, we use the same system, but configured with less memory (8GB), and use a 2-pass sort. It sorts the 100GB dataset in 133.0 (±1.5) seconds, with an average power of 158.2W (±3.4W). It requires 21,042J (±502J), achieving 47,526 (±1135) sorted records per Joule. Compared to the existing (2010) Daytona record, we reduce energy by 25%, and improve records per Joule by 33%. Our system also beats the existing (2010) Indy record, reducing energy 16% and improving records per joule by 19%. (We note that our low-power system, based on an Intel® Atom™ processor, coupled to 4 SSDs performs almost as well (22,361J, 44,720 rec/J); it, too, beats both the Daytona and Indy records.)

Finally, our 1TB (10^{10} records) JouleSort entry once again uses the same setup as the 100GB sort (Intel® Core™ i7 processor, two hardware RAID cards, 16 SSDs, and a boot drive). It sorts the 1TB dataset using two passes in just 1359 seconds (±3.3s) with an average power of 168.3W (±2.9W). It requires 228,817 Joules (±4360J), achieving 43,703 (±833) sorted records per Joule. This reduces energy by 88% and improves sorted records per Joule by 729% compared to the winning 2011 1TB Daytona entry. Compared to the winning 2011 1TB Indy, our entry reduces energy by 61% and improves sorted records per Joule by 151%.

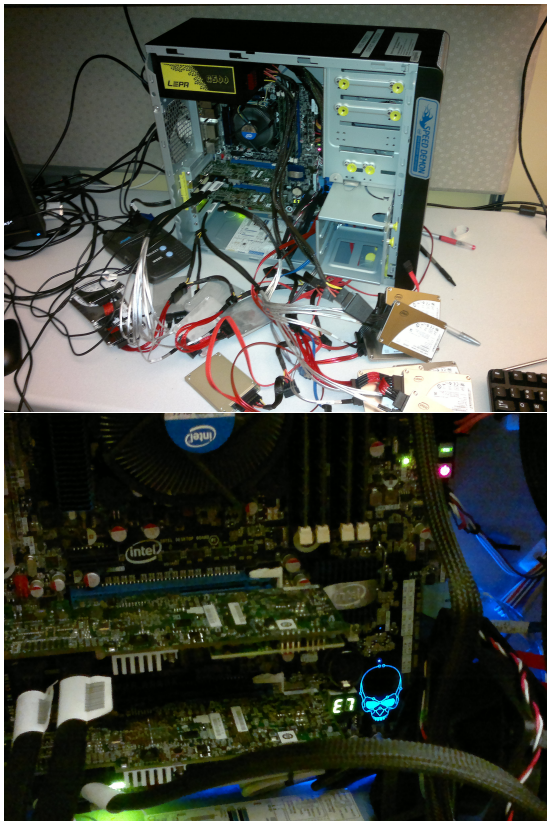

Figure 1: Desktop system with 16 SSDs

2 Hardware

We have tested two systems in various configurations.

Desktop Our large system uses an Intel® Core™ i7-2700K ("Sandy Bridge"), a 3.5 GHz quad-core processor (with hyperthreading, TurboBoost-enabled, 95W TDP) paired with 8–16GB of DDR3-1333 DRAM (2–4x 4GB DIMMs). The mainboard, an Intel® DZ68BC, provides 4x 6-Gb/s SATA, 4x 3-Gb/s SATA, and 1x eSATA ports. Unfortunately, this generous set of ports cannot be pushed to maximum because they all share the DMI v2 bus connection to the processor, which has a theoretical limit of 20Gb/s. To get around this bandwidth bottleneck, we populate the two PCIe "graphics" slots (which provide a total of 16 PCIe 2.0 lanes directly connected to the processor) with two Intel® RS25DB080 hardware RAID cards. These are based on an

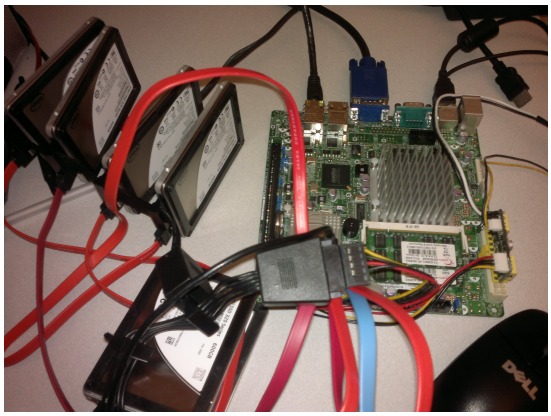

Figure 2: Intel® Atom™ system with 6 SSDs

LSI 2208 chipset, and provide 8x 6Gb/s SAS/SATA ports each.

For storage, we use 16 SATA-based Intel® 710 series SSDs, each with 300 GiB capacity. Eight drives are attached to each RAID card, configured as two RAID-0 sets. An additional boot drive (Intel® SSD, 510 series, 120 GiB) is attached to the motherboard.

The power supply is a Lepa 500G, a Gold 80 Plus rated 500W power supply. The system is in a standard ATX case, with the side removed to allow all of the drives to be connected. For cooling, the power supply has an internal fan, and we use the stock heatsink and fan that comes with the processor. An additional case fan is also used, though we repositioned it to ensure airflow to the RAID cards.

Atom Our smaller system configuration uses an Intel® Atom™ D510 ("Pineview"), a 1.66 GHz dual-core processor (with hyperthreading, 13W TDP) paired with 2–4GB of DDR2-667 DRAM (1–2x 2GB DIMMs). The mainboard, a SuperMicro X7SPA-HF, provides 6x 3-Gb/s SATA ports. For storage, we use between 4–6 SATA-based Intel® 320 series SSDs, each with 600 GiB capacity. A small partition on one of the drives served as the boot disk. The system power supply is a picoPSU power adapter with a 12V, 60W DC power supply. There are no fans in this system.

Both the Desktop and Atom systems use a stock configuration in the BIOS; the only changes were boot options, and enabling AHCI rather than legacy IDE mode for the onboard SATA ports. In particular, no overclocking, voltage tweaking, or fan control options were modified. We also forced both systems to use 100Mbit/s Ethernet (by attaching them to a 100Mbit/s switch); this decision saved approximately 1W over using Gigabit Ethernet.

2.1 System price and power

All of the hardware components are commercially available. Current retail prices, primarily from Newegg.com, for the components of the two systems are provided in Table 1 and Table 2. In both systems, the SSD costs dominate the system costs.

Part	#	Unit Price
Intel Core i7-2700K Desktop Processor	1	$340
Intel BOXDZ68BC ATX Motherboard	1	$220
Intel RS25DB080 RAID PCI-E Card	2	$670
Intel 710 300GB SATA SSD	16	$1280
Intel 510 120GB SATA SSD	1	$230
Kingston 4GB DDR3 1333 SDRAM DIMM	4	$22
LEPA G500-MA 500W Power Supply	1	$120
InWin IW-C583TX.BL ATX Chassis	1	$150*
System Total	1	$22808

(* estimated – unavailable)

Table 1: Price List for Desktop System

Part	#	Unit Price
SuperMicro X7SPA-HF	1	$220
2GB DDR2 667 SODIMM	2	$28
Intel 320 600GB SSD	6	$1200
picoPSU ATX power adapter	1	$30*
60W 12V DC Power Supply	1	$20*
System Total	1	$7526

(* Amazon.com)

Table 2: Price List for Atom System

The desktop system as configured idles at approximately 80W and peaks at around 185W. In practice, we saw a broad range of power numbers while running the sorts, depending on the data throughput achieved. The processor itself is rated as 95W TDP, including the built-in graphics pipeline. The RAID cards each are rated 23W maximum power.

The Intel® Atom™ system idles at 23W, and peaks at around 38W. The processor is rated at 13W TDP, including graphics.

3 Software

All of our experiments are run using Mint Linux 12, 64-bit version, with the kernel upgraded to version 3.2.0. No custom drivers are needed for either system. For the 10GB and 100GB sorts on the desktop system, we simply use one HW RAID set as input and output, and the other as temp space. For the 10GB single pass sort, we use SW RAID to stripe data across both RAID sets to maximize bandwidth. For the Atom system, we use 2 drives as a SW RAID-0 set for temp space, and the remaining 2 or 4 drives in another SW RAID-0 set for the input and output files. Except for the boot partitions, all off the volumes are formatted with XFS filesystems. In addition, we did "break-in" the SSDs prior

```
nsort -processes=8 -memory=15000M
      -method=radix
      -format=size:100
      -field=name:key,size:10,off:0,character
      -key=key
      -statistics
      -file_system=/raid0,direct,
               transfer_size=512K,count=24
      -out_file=/raid0/output,direct,
               transfer_size=64M,count=4
      /raid0/inputdata
```

Figure 3: NSort parameters used for best 10GB sorts

```
nsort -processes=8 -memory=7000M
      -method=radix
      -format=size:100
      -field=name:key,size:10,off:0,character
      -key=key
      -statistics
      -file_system=/raid0,direct,
              transfer_size=256K,count=24
      -out_file=/raid0/output,direct,
              transfer_size=64M,count=12
      -temp_file=/raid1/,direct,
              transfer_size=4M,count=24
      /raid0/inputdata
```

Figure 4: NSort parameters for best 100GB, 1TB sorts

to the results presented here, by writing more than capacity to each dirve. This ensures that the garbage collection at the FTL layer is active, avoiding any artificially high sequential write speeds. For this reason, we do not perform any secure erase operations on the drives.

We use the provided `gensort` utility to create the input data files and use the provided `valsort` to validate our final output file. For the actual sorting, we use a trial version of NSort software (http://www.ordinal.com). Nsort parameters for our best runs are shown in Figures 3 and 4. We note that we tweaked the transfer sizes for the input, temp, and output files for different configurations.

Like previous entries that used NSort to compete for JouleSort [1, 2], we meet the 2012 designation for the Daytona category since NSort is a general sort software package.

4 Measurement

We measure the energy consumption during our sort experiment using a WattsUp Pro .NET power meter ([3]). This meter reads to 0.1W precision, and has a specified accuracy of ±(1.5%+0.3)W. We connect the power meter to our test machine using the onboard USB interface and use publicly available software for the power meter to log the power readings once per second. For each run, our execution script first starts the logging software, waits a few seconds for power measurements to start appearing in the log file, then runs the `nsort` command, waits for the sort to complete, and then terminates the power logging. The script inserts sort start and end messages into the power log file, so correlating the correct power measurements with the experiment is not a problem. Our script uses `/usr/bin/time` to measure and report the actual runtime of NSort.

Using the logs, we calculate the energy consumed by averaging the power values that are measured once per second over the duration of the run and multiplying that average power by the runtime reported by `/usr/bin/time`. We have to be careful in computing the average power over a run, since the initial and final 1-second power measurement intervals may only have the sort benchmarking running for parts of the intervals. We compute average power by discarding the two lowest power measurements of the relevant measurements intervals. For example, for our 8.48s experiment, we use the highest 7 values to average the power, ignoring the two lowest (i.e., first and last) values of the 9 pertinent entries. We use this calculated average power and multiply by the actual runtime of the experiment to calculate the total number of Joules.

5 Results

Our results are summarized in the tables below. The final errors reported include measurement error and averge deviation over five runs.

10GB sort on i7 system, 16GB RAM, 16+1 SSDs

	Time (s)	Power (W)	Energy (J)	SRecs/J
Run 1	8.48	164.9±2.8	1398.5±25.1	71507±1287
Run 2	8.45	162.8±2.7	1375.4±24.8	72705±1311
Run 3	8.46	165.6±2.8	1400.6±25.2	71397±1285
Run 4	8.47	164.7±2.8	1395.1±25.1	71678±1290
Run 5	8.51	164.0±2.8	1395.5±25.1	71658±1290
Avg	8.47±0.03	164.4±3.6	**1393.0±32.1**	**71789±1659**

The statistics reported by Nsort during these runs indicate around 690% CPU utilization, 3800 MB/s, and 2.6s for the input phase, and 770% CPU utilization, 1960 MB/s, and 5.4s for output phase. `/usr/bin/time` reports 0.45s longer total run time than Nsort itself. As mentioned above, we use the reported number from `/usr/bin/time` to calculate the duration of the sort.

100GB sort on i7 system, 8GB RAM, 16+1 SSDs

	Time (s)	Power (W)	Energy (J)	SRecs/J
Run 1	131.1	159.0±2.7	20847±354	47968±814
Run 2	134.4	157.8±2.7	21200±360	47169±801
Run 3	134.4	157.9±2.7	21232±360	47098±799
Run 4	131.2	159.1±2.7	20876±354	47903±813
Run 5	134.1	157.1±2.7	21057±358	47490±807
Avg	133.0±1.5	158.2±3.4	**21042±502**	**47526±1135**

The statistics reported by Nsort during these runs indicate around 700% CPU utilization, 1660 MB/s, and 62s for the input phase, and 560% CPU utilization, 1400 MB/s, and 72s for output phase. `/usr/bin/time` reports around 0.1s longer total run time than Nsort itself. As mentioned above, we use the reported number from `/usr/bin/time` to calculate the duration of the sort.

100GB sort on Atom system, 2GB RAM, 4 SSDs

	Time (s)	Power (W)	Energy (J)	SRecs/J
Run 1	740.2	30.24±0.75	22384±558	44674±1114
Run 2	739.8	30.23±0.75	22367±558	44709±1115
Run 3	739.4	30.25±0.75	22365±558	44712±1115
Run 4	739.9	30.22±0.75	22361±558	44720±1115
Run 5	738.9	30.22±0.75	22329±557	44785±1117
Avg	739.6±0.5	30.23±0.76	**22361±571**	**44720±1141**

The statistics reported by Nsort during these runs indicate around 385% CPU utilization, 268 MB/s, and 375s for the input phase, and 395% CPU utilization, 275 MB/s, and 365s for output phase. `/usr/bin/time` reports around 0.3s longer total run time than Nsort itself. As mentioned above, we use the reported number from `/usr/bin/time` to calculate the duration of the sort.

1TB sort on i7 system, 8GB RAM, 16+1 SSDs

	Time (s)	Power (W)	Energy (J)	SRecs/J
Run 1	1351	168.5±2.8	227606±3821	43936±738
Run 2	1361	168.5±2.8	229321±3850	43607±732
Run 3	1361	168.1±2.8	228732±3841	43719±734
Run 4	1363	168.3±2.8	229492±3853	43575±732
Run 5	1360	168.3±2.8	228935±3844	43681±733
Avg	1359±3.3	168.3±2.9	**228817±4360**	**43704±833**

The statistics reported by Nsort during these runs indicate around 740% CPU utilization, 1575 MB/s, and 635s for the input phase, and 580% CPU utilization, 1385 MB/s, and 725s for output phase. `/usr/bin/time` reports about 0.25s longer total run time than Nsort itself. As mentioned above, we use the reported number from `/usr/bin/time` to calculate the duration of the sort.

All of the results presented here improve on the existing (2010/2011) records for both Daytona and Indy categories in the 10GB, 100GB, and 1 TB JouleSort competitions.

5.1 Additional Results

Tables 3–6 summarize some of our experiments with a broader range of configurations.

Acknowledgments

We would like to thank Intel's Frank Berry and Robert Stoddard for advice on RAID card performance.

	CPU	RAM	Drives
Med Atom	Atom D510	2GB	4x600GB Intel 320
Big Atom	Atom D510	4GB	6x600GB Intel 320
Desktop	i7 2700K	16GB	16x300GB Intel 710 *
Desk 8g	i7 2700K	8GB	16x300GB Intel 710 *

Table 3: System Configurations. *Desktop and Desk 8g are configured with 2 RAID-0 arrays of 8 SSDs each.

	idle (W)	peak (W)	avg (W)	time (s)	energy (kJ)
Med Atom	20.1	32.2	30.7	66.5	2.04±.07
Big Atom	23.3	37.2	35.5	65.8	2.34±.06
Desktop	81.6	170	164.4	8.5	1.39±.03
Desk 8g	77.8	177.9	150.4	14.1	2.13±.04

Table 4: 10GB Sort Results

	idle (W)	peak (W)	avg (W)	time (s)	energy (kJ)
Med Atom	20.1	32.1	30.2	740	22.36±0.57
Big Atom	23.3	36.5	35.3	714	25.19±0.63
Desktop	81.6	174.7	165.7	136.1	22.55±0.47
Desk 8g	77.8	177.9	158.2	133.0	21.04±0.50

Table 5: 100GB Sort Results

	idle (W)	peak (W)	avg (W)	time (s)	energy (kJ)
Big Atom	23.3	35.9	33.9	8911	302.0±7.9
Desktop	81.6	185.0	175.0	1397	244.4±4.5
Desk 8g	77.8	179.6	168.3	1359	228.8±4.4

Table 6: 1TB Sort Results

References

[1] J. D. Davis and S. Rivoire. Building energy-efficient systems for sequential workloads. Technical Report MSR-TR-2010-30, Microsoft Research, Mar. 2010.

[2] S. Rivoire, M. A. Shah, P. Ranganathan, and C. Kozyrakis. JouleSort: A balanced energy-efficient benchmark. In *Proc. ACM SIGMOD*, Beijing, China, June 2007.

[3] WattsUp. .NET Power Meter. http://wattsupmeters.com.