NTOSort

Andreas Ebert
Vienna, Austria
April, 2013

Abstract

In this paper, I present the results of my submission to the 2013 edition of the sort contest. One system sorted 10GB with 889 Joule, or 112545 sorted records /Joule, which represents an improvement of 57% vs. the existing result. For the 100GB Joulesort category, the same system sorted 100GB with 12092 Joule (+74% vs. 2012). A second system executed the 100GB sort with 14292 Joule (+47%) and simultaneously was capable of sorting 100GB in under one minute. The performance of this single socket workstation represents 47% of the 2007 Minutesort record – a cluster with 400 nodes and 2400 disk drives. In the 1TB sort category, a desktop system achieved 168242 Joule (+36%). All systems were built with widely available hardware and used Windows 8 Pro and the sort package nsort.

1. Introduction

This paper summarizes the selection, configuration and results of 3 systems build to improve the existing sorting results for 10GB, 100GB and 1TB Joulesort (Daytona and Indy category). Section 2 describes the hardware selected and the configuration. Section 3 presents the software configuration. Section 4 describes the measurement approach. Section 5 concludes with the Joulesort results.

2. Hardware

Traditionally, the computational capabilities of hardware improved in the last years faster than the rate of progress we saw in associated I/O performance. With the emerging big data era, the last years showed promising developments in improved I/O capabilities. The move of the PCI Express root complex directly on the CPU provides significant reduction in latency and an increase in bandwidth. The second development supporting higher I/O capabilities can be seen in the new generation of fully SSD capable and cost effective RAID controllers. The design of previous generation RAID controllers were built on the I/O patterns and capabilities of hard disk drives. Until very recently, RAID controllers were either often a limiting factor for parallel SSD performance or not cost effective. The new generation supports the PCI Express 3.0 protocol currently mostly in x8 configurations.

The third interesting development for highly parallel I/O projects can be seen in recent generation of cost effective consumer SSD with high sustainable performance, especially with regards to sustainable write performance and pro-active garbage collection.
In a different configuration not optimized for Joulesort, a dual socket Windows 8 workstation is able to achieve over 20 GB/s sequential disk transfer rates (measured with IOMeter). It is also within the reach of increasingly more users to utilize cost effective random I/O with 2.2 mio IOPS and 4 KB sectors (8.8 GByte/s) , packaged in basically a classic PC tower case. Two of the systems described in this paper are built with some of these concepts in mind.

Configurations

Notebook: *Lenovo X220 with docking station (Joulesort 10GB and 100GB entry)*
Usually notebooks provide a rather limited I/O capability, yet they are very energy efficient. An available Lenovo W520 portable workstation could not saturate the 4-core mobile CPU with the available connections for drives. Modern ultrabooks are built with other design priorities and are less effective with regards to high I/O loads. A Lenovo X220 portable, attached to the ThinkPad UltraBase Series3 docking station provided a good balance of I/O, power consumption & compute power and was selected for the 10GB and 100GB Joulesort submission. The CPU is a

dual core 2.8 GHz Intel i5-2640M. With the flip-on docking station an energy efficient I/O extension is possible. The connector of the Lenovo power supply was connected to the notebook. No additional external power connection for the docking stations is necessary as it is supplied directly from the notebook. The system was configured for both categories with 16 GB main memory 16GB (2 x Kingston SoDimms). The SATA ports in the notebook and docking station provide SATA III support (6 GBit/s). An OCZ 120GB mSATA Nocti drive served as the system drive and 2 Samsung 840 Pro 256 GB drives were used as data drives. One SSD replaced the standard hard disk of the notebook and the second SSD replaced the DVD drive in the docking station. Both SSDs had 6 GBit/s SATA connectivity.

After initial experimentation with 15 different SSDs, the Samsung 840 Pro SSDs were selected based on 3 factors: 1) Almost symmetrical read and write speed, 2) very high read/write performance topping 500MB/s and 3) predictable performance in long write operations without significant performance loss with high fill factors – which happens with sort data sets relatively frequent.

No modifications were done in the BIOS and no further rather rare optimizations like forcing the LAN port to 100 Mbit/s operation.

Joulesort configurations for desktops and servers do not include the energy consumed for the display. To compensate this energy disadvantage for the notebook, the display was switched off and remote desktop software was used to control the notebook remotely.

Workstation: *Single Xeon E5-2687 (100GB Joulesort entry)*

The second system represents the current performance category for single socket systems. The Intel E5-2687W CPU is an 8-core (16 with HT) design with 3.1 GHz base frequency. Four memory channels and 20 MB cache provide the base for higher I/O workloads. The system described in this paper is based on an ASUS Z9PE-D16 dual socket server motherboard, but only half populated. This motherboard was extracted from the before mentioned workstation and reconfigured for the 100 GB Joulesort run. Available desktop motherboards are limited to max 64 GB, but the respective Xeon based systems allow up to 384 GB main memory per socket in either single or dual socket systems. This motherboard was populated with one Intel Xeon E5-2687W CPU and 128 GB main memory (8 x 16 GB ECC Ram)

To save energy, only one host bus adapter was used. Adaptec provides a new generation of adapters supporting PCI Express 3.0 and 16 SATA III ports and this HBA fits with 16x Samsung 840 Pro 256 GB SSDs to this very specific workload well. With a total capacity of 2 TB the smaller 128 GB SSD versions would be large enough from a capacity perspective, but this choice was made on the transfer symmetry the 256 GB version provides. In this configuration with one 16 SSD stripe set, IOMeter measures 6.8 GB/s read transfer rates and 6.6 GB/s write transfer rates.

Power is supplied with a "be quiet!" Straight Power E9 500W PSU, rated with 90+% efficiency and sufficiently powerful to drive a half populated system with one CPU.

Desktop: *Intel i7-3770K (1 TB Joulesort entry)*

Leveraging some components of other systems used, this system builds on the 4 TB capacity of the 16 SSDs connected to one controller. To adapt to the lower compute performance requirements for a 2-pass sort (necessary for the 1 TB input file), a small LGA-1155 based system was used. The 16 SSDs connected to the one Adaptec controller were split in 2 HW RAID0 configurations. This system configuration is rather memory bandwidth limited than I/O limited. Each RAID0 volume has a theoretical transfer rate of more than 4.000 MB/s. 60% of this maximum rate was used in the second pass, limited by the main memory bandwidth of the LGA-1155 socket. This system has only 2 memory channels with a total memory bandwidth of 16-18 GB/s. Depending on the bandwidth needs of the CPU, between 30% and 35% of this bandwidth is available for I/O.

System price and power

All hardware is or was commercially available, as some of the parts used for this entry are now discontinued.

Part	#	Unit price	Item price	% syscost
Lenovo X220 4291 (* discont., ex-MSRP)	1	$1,920	$1,920	64.8%
Thinkpad Ultrabase Series 3	1	$160	$160	5.4%
Thinkpad Ultrabay Adapter III (* est)	1	$60	$60	2.0%
Kingston 8GB DDR SoDimm	2	$60	$120	4.0%
Samsung 840 Pro 256GB	2	$250	$500	16.9%
OCZ Noct 120 GB mSATA	1	$205	$205	6.9%
System total			$2,965	

Table 1: Price List for Notebook System (JouleSort 10GB, 100GB)

Part	#	Unit price	Item price	% syscost
Intel Xeon E5-2687W CPU	1	$1,935	$1,935	23.0%
ASUS P9ZE-D16 Motherboard	1	$480	$480	5.7%
Kingston 16 GB DDR3 ECC 1600MHz	8	$135	$1,080	12.8%
Samsung 840 Pro 256GB	16	$250	$4,000	47.6%
Samsung 840 Pro 128 GB	1	$125	$125	1.5%
Adaptec HBA 71605E	1	$400	$400	4.8%
Adaptec cables 2279800-R	4	$25	$100	1.2%
Sharkoon case Rebel 12 (est)	1	$120	$120	1.4%
PSU be quiet! Straight Power E9 500Watt	1	$165	$165	2.0%
System Total			$8,405	

Table 2: Price List for Workstation (JouleSort 100GB)

Part	#	Unit price	Item price	% syscost
Intel i7-3770K desktop processor	1	$330	$330	6.1%
ASUS P8H61-MX USB3	1	$160	$160	2.9%
Corsair 8GB DDR3 RAM	2	$60	$120	2.2%
Samsung 840 Pro 256GB	16	$250	$4,000	73.4%
Samsung 840 Pro 128 GB	1	$125	$125	2.3%
Adaptec HBA 71605E	1	$400	$400	7.3%
Adaptec cables 2279800-R	4	$25	$100	1.8%
PSU be quiet! Straight Power E9 500Watt	1	$165	$165	3.0%
Case	1	$50	$50	0.9%
System total			$5,450	

Table 3: Price List for Desktop System (JouleSort 1TB)

3. Software

All results of systems submitted are running Windows 8 Pro 64-bit. Starting with a fresh installation, all security updates and patches were applied before measuring sort performance. The driver for the Adaptec 71605E HBA is available from the manufacturer's website.

Sorting is processed with the `nsort` application by www.ordinal.com . I used the provided `gensort` utility to generate the input files and validated all sorted files with `valsort`.

To support advanced functionality in `nsort`, all operating systems were configured to support `Lock pages in memory` and `Perform volume maintenance tasks`.

To enable (admin privileges are required):
WindowsKey+R, start GPedit.msc
Select Windows Settings -> Security Settings -> Local Policies -> User Rights Assignment
Add the current user to the list of authorized users for the two options

SSDs are often exhibiting different performance characteristics depending on usage patterns and previous secure erasure. All SSDs used in the runs are well used SSDs with many TB of previous write activities and stable write performance. No secure erase was applied during the project.

Configurations

10 GB and 100GB Joulesort / Notebook
System disk is the OCZ Noct mSATA SSD. Both Samsung SSDs are configured as one software-based RAID0 filesystem to allow the fastest possible transfer rates for a single pass sort. Main memory is 16 GB. To align the setup of the notebook with the desktop setups, the built in screen was set to screen saving mode. The system was operated with Remote Desktop Services. Power mode was set via the operating system mechanisms available in Windows 8 Pro to "power saving" with no further modifications. Idle power consumption is 7.6 W. Due to the high power usage during sorts, the notebook enters throttling mode when connected to the 65 Watt external power supply. Connecting it to the default 90 W power supply alleviates this issue. I removed the battery to avoid influencing the power reading through charge/discharge cycles.

Nsort parameters for 10GB sorts

```
-processes=4
-memory=13500M
-method=radix
-touch
-format=size:100
-field=name:key,size:10,off:0,character
-key=key
-statistics
-in_file=d:\ns.dat,direct,transfer_size=16M,
count=64
-out_file=d:\nd.dat,direct,transfer_size=128M,
count=8, preallocate
```

`Nsort` parameters for 100GB sorts. Please note the oversubscription of processes vs. available cores to improve overall CPU utilization in the 2 pass sorts.

```
-processes=6
-memory=12000M
-method=radix
-touch
-format=size:100
-field=name:key,size:10,off:0,character
-key=key
-statistics
-in_file=d:\ns.dat,direct,transfer_size=16M,
count=64
-out_file=d:\nd.dat,direct,transfer_size=128M,
count=8, preallocate
-temp=d:\,direct,transfer_size=64M, count=32,
preallocate
```

100 GB Joulesort / Workstation

Windows 8 Pro was installed on a separate 128 GB SSD. To reduce load on the CPU, one single RAID0 array for the 16 data SSDs was created in the controller. Windows 8 utilizes this as one simple volume. As this configuration with one 8-core CPU (16 with HT) is CPU bound, power mode was set to high performance in the operating system. Read performance is limited not by the speed of the 16 SSDs to the controller, but the limitation of the PCI Express 3.0 x8 interconnect, with a maximum theoretical transfer rate of 8 GB/s. In this configuration up to 6.5 GB/s read transfers are possible. Higher read performance with up to 8 GB/s was reported by `nsort` when the 16 SSDs were connected via 2 controllers.

Nsort parameters for 100GB sorts

```
-processes=16
-memory=124000M
-method=radix
-touch
-format=size:100
-field=name:key,size:10,off:0,character
-key=key
-statistics
-in_file=d:\ns.dat,direct,transfer_size=16M,
count=64
-out_file=d:\nd.dat,direct,transfer_size=64M,
count=64, preallocate
```

1 TB Joulesort /Desktop

The 16 SSDs and the Adaptec controller used in the 100GB workstation sort were reused for this configuration. The 16 drives were split in 2 RAID0 arrays provided by the controller. Windows 8 accessed them as two logical drives. Power mode for the i7-3770K CPU was set in the operating system to "high performance".

```
-processes=12
-memory=13500M
-touch
-method=radix
-format=size:100
-field=name:key,size:10,off:0,character
-key=key
-statistics
-in_file=d:\ns.dat,direct,transfer_size=16M,
count=32
-out_file=d:\nd.dat,direct,transfer_size=128M,
count=8, preallocate
-temp=e:\, direct,transfer_size=64M, count=16,
preallocate
```

NSort configuration options

A few comments on the configuration options selected for the `nsort` software package:

"`-touch`" option

This option is crucial for the workstation single sort pass. All memory for the 100 GB input data set plus working space has to be allocated in main memory before the input phase can start. "`-touch`" parallelizes the memory allocation to the available cores.

"`processes`"

For all single pass sorts, the number of processes `nsort` used was set to the actual number of logical cores in the system. Logical cores = cores as seen by the task scheduler of the OS, independent if real cores or SMT cores (in Intel language it is called hyperthreading).

For all two-pass sorts (100GB on the notebook and 1TB on the desktop system), the number of processes was set 50% higher than the number of logical cores in the system. For the 100GB sort run with the notebook, `nsort` used 6 threads on a system with 4 cores, and in the case with the desktop system with 8 cores, the number for `nsort` was set to 12. This "oversubscribed" setting increased the overall performance vs. a setting based on the number of cores in the system. Most likely, this was driven by a better fill ratio in the memory controller pipeline of the CPUs - of threads waiting for its data to be read from or written to main memory.

Buffer sizes for input and output files

The performance sensitivity of buffer sizes is much more pronounced for output files. For output, large buffer sizes with fewer buffers were used, for input files, smaller buffer sizes with more buffers were configured.

Total memory allocated to buffers

1) Notebook (10GB single-pass run): 1 GB in, 1 GB out
2) Notebook (100GB two-pass run): 1 GB in, 1 GB, out, 2 GB temp
3) Workstation (100GB single-pass run): 1GB in, 4 GB out
4) Desktop (1TB two-pass run): 512MB in, 1 GB out, 1 GB temp

4. Measurements

I measured the energy consumption during this experiment using a WattsUp! .NET power meter. According to the manual, this meter reads to a precision of 0.1 W and has a specified accuracy of ±(1.5%+0.3) W.

All systems under observation were connected to the power meter via the onboard USB interface. Logging of power data was provided by the WattsUp! USB data logging software, downloaded from WattsUp's website. At time of testing there was no readily available script utility available and given the short amount of time to conduct the tests (one

weekend) a combination of the measurement approaches as described in the Psort and Fawnsort papers was applied.

1) The WattsUp! data logger was set to one second measurement intervals.
2) The system under test was in idle mode
3) The start of the logging was done before the execution script was started.
4) Reflecting the approach of the PSort team, the execution script waits for 2 seconds, creates a energy signature in the power log, wait for another 2 seconds and starts the sort application. The command line utility used for creating the energy spike was Intel's publically available `Linpack` benchmark. Set to a dimension of 2000, `Linpack` created this easy to identify energy signature in the log file. `Linpack` creates a fast rising and a fast falling edge in the power log. The falling edge (moving from high energy consumption during `Linpack` back to the idle power level is more pronounced and used as trigger.
5) The power readings of the 2 seconds (wait time of the script) after `Linpack` are discarded.
6) The number of seconds the test run lasted is taken to extract the number of power measurements the WattsUp! logging utility recorded during the test run.

Extracting the power log data per run was done on the principles described by the Fawnsort team.

1) Exclude the first and last measurement points for potential fractional readings.
2) Calculate the energy consumed by averaging the power values measured once per second in the log as described in (1) by the total time reported by the external time command.

The external timing utility used is the `timethis` command from the former Windows Resource Kit.

5. Results

My results for the different categories are listed in the tables below. The final numbers include average deviation over five runs. The new rules require the publication of result with skewed input data sets. All configurations are well within the allotted time. CRC codes are listed for normal data sets, for skewed data sets the CRC code and the number of double records are listed.

10 GB JouleSort, notebook, 16 GB RAM, 2+1 SSDs

	Time(s)	Power(W)	Energy(J)	Srec/J	MaxPwr(W)
Run 1	26.5	33.2	878	113864	49.6
Run 2	25.8	34.8	895	111697	49.8
Run 3	26.3	34.1	897	111512	50.3
Run 4	27.3	32.7	892	112116	49.5
Run 5	25.8	34.1	880	113578	49.2
Avg	**26.3**	**33.8**	**888.5**	**112545**	
stdev	0.6	0.8	8.6	1092.6	
Runtime skewed	28.6	33.6	960	104186	108.5%

CRC sort	2faf0ab746e89a8
CRC skewed	2fae307aff171ad
Doubles skewed	251515

`Nsort` reports for this single pass sort around 240% CPU utilization, 1100 MB/s and 12.3s for the input phase and 270% CPU utilization, 905 MB/s and 12.1s for the output phase. `Timethis` reports a 1.9s longer time than `nsort` itself. I used the time reported by `timethis`. Please note, that the system price of this entrant represents approx. 13% of last year's configuration.

100 GB JouleSort, notebook, 16 GB RAM, 2+1 SSDs

	Time(s)	Power(W)	Energy(J)	Srec/J	MaxPwr(W)
Run 1	395.6	30.4	12,042	83040	45.0
Run 2	406.3	29.9	12,156	82262	43.7
Run 3	400.0	30.5	12,190	82033	43.7
Run 4	396.9	30.4	12,080	82783	45.3
Run 5	396.0	30.3	11,993	83381	45.3
Avg	**399.0**	**30.3**	**12,092**	**82697**	
stdev	4.4	0.2	80.9	553.0	
Runtime skewed	409.4	34.6	14,171	70566	102.6%

CRC sort	1dcd615efb9dfe11
CRC skewed	1dcd5b1360662446
Doubles skewed	25471511

`Nsort` reports for this 2-pass sort a CPU utilization of 270%, 507 MB/s and 195.5s for the input phase and 260% CPU utilization, 504 MB/s and 201.7s for the output phase. `Timethis` reports a 1.8 sec longer time than `nsort` itself. I used the time reported by `timethis`.

100 GB JouleSort, workstation, 128 GB RAM, 16+1 SSDs

	Time(s)	Power(W)	Energy(J)	Srec/J	MaxPwr(W)
Run 1	58.2	246.9	14359	69641	283.3
Run 2	58.5	244.8	14320	69831	284.2
Run 3	58.3	244.3	14253	70163	284.5
Run 4	58.8	242.2	14237	70238	283.0
Run 5	58.5	244.2	14292	69967	284.2
Avg	**58.5**	**244.5**	**14292**	**69967**	
stdev	0.2	1.7	49.7	243.1	
Run Sk	66.3	242.6	16075	62207	113.4%

CRC sort	1dcd615efb9dfe11
CRC skewed	1dcd5b1360662446
Doubles skewed	25471511

`Nsort` reports for this single pass sort, 920% CPU utilization, 6350 MB/s and 23.1s for the input phase and 1475% CPU utilization, 3550 MB/s and 30.4s for the output phase. `Timethis` reports a 5.0 sec longer time than `nsort` itself. I used the time reported by `timethis`.
It is interesting to compare a single system with previous minutesort systems. Sorting 100 GB/min in a desk workstation represent approx. 47% of the 400 node cluster result in 2007 or 20% of the amount of data sorted by the 1406 node cluster in 2009. While this informal comparison is obviously skewed by the timeline involved, it might show the potential on increased research about efficient utilization of single systems complementing ongoing research on scaling.

1TB JouleSort, i7-3770K desktop, 16 GB RAM, 16+1 SSDs

	Time(s)	Power(W)	Energy(J)	Srec/J	MaxPwr(W)
Run 1	1412.3	117.9	166541	60045	137.1
Run 2	1418.1	117.6	166825	59943	137.2
Run 3	1475.7	116.0	171110	58442	137.1
Run 4	1455.2	115.9	168605	59310	137.1
Run 5	1425.2	118.0	168127	59479	137.4
Avg	**1437.3**	**117.1**	**168242**	**59444**	
stdev	27.1	1.1	1821.5	639.1	
Runtime skewed	1598.1	114.2	182547		111.2%

CRC sort	12a06cd06eeb64b16
CRC skewed	12a068878c44b73f2
Doubles skewed	886499324

`Nsort` reports for this 2-pass sort run 640% CPU utilization, 1580 MB/s and 644.5s for the input phase and 435% CPU utilization, 2300 MB/s and 792.1s for the output phase. . `Timethis` reports a 0.7 sec longer time than `nsort` itself. I used the time reported by `timethis`. Please note the relatively higher variability of the individual runs. Like all other runs, these runs are taken in sequential order. Additionally, the 2x RAID0 setup with 2 TB each filled the source volume to more than 97% of total capacity. Most likely this increased the impact of unsynchronized garbage collection of the 8 SSDs in one RAID volume on the sort run times.

Suggestion

Recognizing the rapidly evolving world of IT technologies, I like to suggest to the community a proposal for discussion:
The raise of cloud computing allows interesting new perspectives currently only partially addressed in the existing sort categories. The cloud economic model and inclusion of operational costs doesn't fit perfectly to the current Joulesort or Pennysort categories.

The lack of capital investment needed would give many more interested researchers access to computing resources to further research on sorting, along the lines of "How much data can be sorted in $5 worth of fully loaded cloud compute time ?"

Personal note

While this report is a personal project, I'd like to share my professional affiliation with Microsoft Corporation, currently as the Regional Technology Officer for Western Europe. In my free time, I'd like to investigate all things digital, be it big back-end systems, new devices or HPC or big data computing. Somehow naturally, I landed in "sort land". I'd like to thank Chris Nyberg for granting access to nsort, powering many of the past and this submission.