SheenkSort: 2003 Performance / Price Sort and PennySort

Lei Yang, Hui Huang, Zheng Wan, Tao Song

March 30th, 2003

SheenkSort: 2003 Performance / Price Sort and PennySort

Lei Yang[1], Hui Huang[2], Zheng Wan[3], Tao Song[4]

yanglei@sheenk.com hansen@sheenk.com cos@sheenk.com songtao@sheenk.com

Abstract: The technology of sorting is one of the most fundamental and important technologies in computer science. The external sorting deals with various operations including disk I/O, memory management and CPU-burdened computation, thus has been widely accepted as an overall benchmark to evaluate the processing power of computers[5]. Among such benchmarks the PennySort and the Performance / Price are both aiming at the highest cost efficiency. SheenkSort[6], with the new YHSort Framework[7] fully considering the statistic properties of the data to be processed, and with carefully designed system architecture, exploits much deeper the potential of popular desktop PC than before. It is able to sort 42.28 GB[8] data (454,033,408 records of 100 bytes each) for a penny, which is quite over four times of the last year's Daytona PennySort record setup by THSort (9.8GB data), or about three and a half times of the last year's Indy PennySort record setup by DMSort (12.2GB data). This paper presents the main considerations for SheenkSort and reports the results for PennySort, Performance / Price Sort, as well as Datamation Sort and Minute Sort.

2003 Daytona & Indy PennySort Result

Hardware Considerations

Hardware components of SheenkSort may be all purchased at http://www.ussa.com. With the help of the new YHSort Framework CPU is unburdened and the AMD Athlon XP 1700 is found to be powerful but cheap enough to meet our request. When considering the motherboard which is the most important among all parts, the new nForce-2 chipset (SPP + MCP) attracts us after

[1] State Key Laboratory of Intelligent Technology and Systems, Department of Computer Science & Technology, Tsinghua University, Beijing, China
[2] NSFOCUS Information Technology Co., Ltd., Beijing, China
[3] Department of Physics, Tsinghua University, Beijing, China
[4] Institute of Computer Network Technology, Department of Computer Science & Technology, Tsinghua University, Beijing, China
[5] Sort Benchmark homepage at http://research.microsoft.com/barc/SortBenchmark/.
[6] Named after the team.
[7] Named after the designers, Yang, L. and Huang, H. The YHSort Framework includes both the External YHSort and the Internal YHSort algorithms. The key idea is to minimize the number of comparing and exchanging operations between records by estimating the target position of each record using statistic information. The YHSort Framework uses a general method to gather this statistic information at running-time and it is able to handle all popular cases (including those of non-uniform distributions). The lengthen details of YHSort are not included in this paper. Contact the authors if this interests you.
[8] Throughout this paper KB = 1024 bytes, MB = 1024^2 = 1048576 bytes, and GB = 1024^3 = 1073741824 bytes.

careful comparison. It shows excellent performances of both IDE and memory I/O. Unfortunately the nForce-2 chipset is still not widely supported until today, and the MSI K7N2-L becomes our choice. The nForce-2 has a 128-bit DDR memory channel which helps a lot the internal sorting as one of the fundamental procedures of the external sort. PC2700 had been considered and was shown to benefit this internal sorting procedure, but that required an expensive AMD Athlon XP 2600+ which cost us almost $200 more than the economic Athlon XP 1700. That is the reason we choose PC2100. The choice of hard disk drive is also very import since the external sorting is almost fully disk-I/O-bounded. After the comparison of various brands of various capacities, the Maxtor DiamondMax Plus 8 of 40GB (7200RPM, ATA133) outperformed the others, whose throughput reaches about 50MBps in the test of stand-alone sequentially reading or writing, and the overall throughput of all the four hard disks is as high as 125MBps at running time.

Operation System

Linux is chosen by SheenkSort not only because it has high performance but also because it is open. More OS implementation details may be discovered by analyzing the source code, and it is also possible to inject codes into OS to monitor the behaviors of both the hardware and software to help us design better system (the monitoring codes are just for debugging purpose). Out of so many versions of Linux, the version of Redhat 7.2 is selected simply because we happened to have a copy at hand. You may choose any version, even a self-made-up one if you would like.

Kernel

Generally later kernel may have more functions and higher performance. But unfortunately for the new-born nForce-2 MCP chipset, a serious bug is found in the latest kernel version that ATA133 is forced as ATA33. This bug might have been fixed by patching the kernel, but considering generality we give it up and select the kernel version of 2.4.20.

File System

In a practical Linux environment the ReiserFS is known for its efficiency and stability. But the Ext2FS shows higher performance when handling huge files. The shortage of Ext2FS is that a long period of about 25 seconds has to be spent on deleting a 20GB file. As the tradeoff, XFS is selected because it can handle huge files on both stand-along disks and disk raid efficiently enough, and these huge files can also be deleted within short period.

Programming considerations

Goal of programming is the highest system performance. For SheenkSort this will never be achieved unless all hardware components work smoothly under high pressures to exploit the potentials, and the loads of all components are balanced so that they cooperate well with each other. SheenkSort takes parallelism (reading, writing & computing) using multi-thread instead of multi-process though multi-process is more efficient in Linux. The simpler mechanisms for

synchronizing, resource sharing and efficiently scheduling between threads remedy its shortage. The disk seeking overhead was cited as a prime bottleneck in DHSort[9]. Analysis shows the number of seeking overhead will be no less than $(N / R)^2$ in theory, where N is the data size (42.28GB for SheenkSort) and R the maximal data size that can be handled by internal sorting algorithm which is normally no greater than a third of the physical memory in a parallel system where reading, sorting and writing happen at the same time. The seeking overheads are fully considered by the External YHSort algorithm, and the number of seeking operations is very close to that limit. It may be puzzling that the two disks storing the temporary data do not make up a raid in SheenkSort. Analysis shows that raid 0 doubles the number of seeking overheads since each cluster of data has to be split into two to store in different disks. On the other hand, since the seeking overheads are unavoidable, the External YHSort succeeds in shifting them to the CPU- and memory-burdened sorting-pass such that the system bus will not have nothing to do when the reading thread is seeking.

Internal Sorting algorithm

The internal sorting algorithm had never been a serious problem until today when the disk I/O is exploited to such an extent as in SheenkSort. In the sorting pass about 60MB data per second has to be processed by the sorting thread while about 60% CPU time is taken up by disk I/O operations. Fortunately the Internal YHSort algorithm whose goal is to minimize the number of comparing & exchanging operations succeeds in handling that huge amount of data without too much CPU time. Imbedded ASM codes including some MMX instructions are used to optimize this internally-sorting algorithm.

Bottleneck

Four hard disks are working concurrently in SheenkSort, while the overall disk throughput is much less than the total transferring rate of each disk when working along. This limit of disk I/O comes from the MCP chipset, and is hoped to be relieved in the future. What surprises us more is the delay between non-sequential memory accesses. This delay degrades the memory performance and thus the overall system performance very much although the bandwidth of the PC2100 memory bus is rather wide.

Hardware Details

All parts of SheenkSort may be purchased at http://www.ussa.com. Details are listed in Table 1, where the SKU is a serial number which may help locating the item on that web site:

[9] Helmkamp, B., McCready K.. 1999 performance/price sort and pennysort. http://research.microsoft.com/barc/SortBenchmark/.

System		SheenkSort			
Qty.	**Component**	**Description**	**SKU**	**Price**	**Cost**
1	CPU	AMD ATHLON XP1700 RETAIL BOX CPU	101489	$67.00	$67.00
1	Motherboard	MSI K7N2-L NFORCE2 K7 MAINBOARD[10]	103054	$87.00	$87.00
2	Memory	KINGSTON DDR 2100 KVR266X64C25/512 512MB RETAIL	103111	$66.00	$132.00
4	Hard Disk Drive	MAXTOR 40GB HDD RPM7200 ATA133 OEM	101462	$64.00	$256.00
1	Video Card	JATON TRIDENT 9750PCI 4MB 3D VGA NO TV VIDEO-67Pro	102341	$18.00	$18.00
1	Case	FIC Mid Tower 300W ATX CASE with Front USB/90 days	102704	$19.00	$19.00
1	Assembly Fee			$35.00	$35.00
Total Price					$614.00
Time Budget		94,608,000 seconds / $614.00 = **1540.8** second per penny			

Table 1: Prices for SheenkSort Hardware System[11]

The ratios of various component costs are shown in Figure 1.

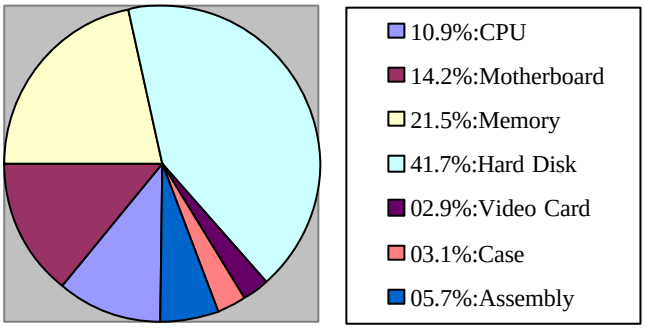

Figure 1: Ratios of various component costs.

It is rather interesting when looking at the prices of last year's winning systems, see Table 2 and Table 3. The total price of SheenkSort is just below that of DMSort a year ago, leading to nearly the same budget times, but the hardware components of SheenkSort are much powerful. On the other hand both AMD Athlon 1700 and two 512M DDR PC2100 are included in both SheenkSort today and THSort a year ago, and even the number and the capacities of hard disk drives are the same. But these same hardware components lead to different prices: the price of SheenkSort is only 71% of the price of THSort, and the budget time for SheenkSort is 436 seconds longer.

[10] With an onboard 10M/100M Ethernet adapter.
[11] All prices were from http://www.ussa.com on March 30, 2003.

System	DMSort (winner of 2002 Indy PennySort)			
Qty.	Component	Description	Cost	%Cost
1	Barebones System	ASUS A7V133 RAID Motherboard AMD Athlon 1GHz Floppy Drive	$306.00	45.51%
3	Memory Chips	256MB PC133	$94	13.96%
2	Hard Disk Drives	60GB IBM Deskstar 60 GXP	$246	36.59%
Total Price		$672.38		
Budget Times		1406 seconds		

Table 2: Main Prices of DMSort (winner of 2002 Indy PennySort)[12]

System	THSort (winner of 2002 Daytona PennySort)			
Qty.	Component	Description	Cost	%Cost
1	CPU	AMD Athlon XP 1700	$135.00	12.20%
1	Motherboard	Abit KR7A-Raid	$117.00	13.65%
4	Hard Disk Drive	IBM 40G ATA100 IDE 7200rpm	$284.00	33.14%
2	Memory	Generic PC2100 512MB DDR for via Chipset	$220.00	25.6%
Total Price		$857.00		
Budget Time		1104 seconds		

Table 3: Main Prices of THSort (winner of 2002 Daytona PennySort)[13]

PennySort Result

The SortGen[14] generate exact 100-byte record under Linux System, making up our data set. Table 4 lists the PennySort result for SheenkSort.

Product	Time Budget	Best Time	Sys	User	Total CPU Time	Sorted GB	Category
SheenkSort	1540.8s	1527.076	671.53	665.67	1337.20	42.28	

Table 4: PennySort result for SheenkSort

Datamation Sort Result

As the origin of sort benchmark, Datamation Sort aimed to seek the fastest way to sort 1M record (100MB data) without considering the cost. As time went by, the time needed by Datamation Sort dropped sharply from 980 seconds (in 1987) to 0.44 second (in 2001). It is so easy to sort 1

[12] Not all components are listed. D, Aaron., M. Alex. DMSort: A PennySort and Performance/Price Sort. http://research.microsoft.com/barc/SortBenchmark/DMsort.pdf.
[13] Not all components are listed. Liu, P., Shi, Y., Li, Z. 2002 Performance / Price Sort and PennySort. http://research.microsoft.com/barc/SortBenchmark/THsort.pdf.
[14] http://research.microsoft.com/barc/SortBenchmark/SortGen.zip.

million records today that Datamation Sort is no longer suitable to be a benchmark, and has been deprecated[15]. TeraByte Sort[16] may be regarded as an alternation for Datamation Sort, though many factors to be considered are different between them. For those PennySort winner it is still worthy of presenting the Datamation Sort results to show how fast they process transactions, because it is still impossible for most of them to handle 1TB data. In practice all the 1 million records may easily be fit in memory and it takes the Internal YHSort algorithm no more than one second to reorder them, with all else have to do are just to read the original data in and to write the sorted data out. But considering generality we insist the original two-pass External YHSort algorithm without changing any of the parameters optimized for PennySort. The Datamation result for SheenkSort turns out to be 4.0 seconds.

Minute Sort Result

Sort as many 100-byte records as you can in one minute for this benchmark. Like those in Datamation and TeraByte Sort, the Minute Sort winners were also supercomputers since prices are beyond consideration. SheenkSort as a popular desktop PC is able to sort 15728640 records (1.46GB data) in 60 seconds.

2003 Performance / Price Sort Result

It has long been worried that the budget time will keeps growing as the price of the popular desktop PC keeps dropping. Being the ancestor of PennySort, the DollarSort benchmark was discarded when the budget time for a dollar was found too long. It is hard to predict the computers tomorrow, and who knows the typical price at that time? Fortunately the budge time for SheenkSort is still within our patience.

Worrying about that the budget time might be to long in the near future, the PennySort benchmark is being considered to be revised to the Performance / Price Sort benchmark, with the budget time fixed to be one minute. But is such period reasonable, or how long the period will be reasonable? What's more, the sorting time grows faster than linearly as the data size increases which makes a simple division by price not so reasonable, and we prefer the direct comparison between the sorted data sizes.

Another argument is that the super computers are kept out of PennySort because the budget time will be too short for them and they get poor results in PennySort. But the goal of PennySort is to find a resolution not only efficient but also economic. For a system too expensive to afford, nothing will be meaningful. I'm so glad to see that the popular desktop PC is the hero of PennySort.

---

[15] http://research.microsoft.com/barc/SortBenchmark/.

[16] You are required to sort 1 TB data (1024^4 bytes) in this benchmark. Price is not considered as well as Datamation. http://research.microsoft.com/barc/SortBenchmark/.

Nevertheless the Performance / Price results of SheenkSort, together with the historical ones are listed in Table 5 which is calculated in this way[17]:

(1) Sort the largest file (GB) you can in a minute.

(2) Compute the system price per minute (3-year depreciation => system price ($) / 1576800 (min)).[18]

(3) Compute the GB/$ sorted by dividing item 1 by item 2.

Year	MB/sec	GB/$	Systm	Sys Price(M$)	CPU(s)	Category
2003	**25.0**	**3761.8**	**SheenkSort**	**0.000614**	**1**	
2002	8.64	1165.7	DMSort	0.000672	1	Penny/Indy
2002	10	1079.5	THSort	0.000857	1	Penny/Daytona
2001	6.50	608.86	HMSort	0.0010	1	Penny/Indy
2000	6.50	608.86	HMSort	0.0010	1	Penny/Indy
1999	2.23	174.99	Postman Sort	0.0012	1	Penny/Daytona
1999	2.46	220.59	NTSort	0.0010	1	Penny/Indy
1999	3.51	314.51	HMSort	0.0010	1	Penny/Indy
1999	3.78	338.17	HMSort Post-April 1st	0.0010	1	Penny/Indy
1998	1.74	125.00	PostmanSort	0.0013	1	Penny/Daytona
1998	1.74	144.00	NTSort	0.0012	1	Penny/Indy
1997	140.17	8.41	Now 95 , Arpaci-Dusseau	2.0	95	Minute/Indy
1997	86.21	6.27	SGI/Ordinal, Nyberg	1.3	14	Minute/Daytona
1996	100.00	15.76	NOW, Arpaci-Dusseau	0.6	32	Minute/Indy
1995	28.57	2.70	SGI/Ordinal, Nyberg	1.0	16	Minute/Daytona
1995	19.61	37.10	IBM, Agarwal	0.05	1	Minute/Indy
1994	1.72	0.16	IPSC/Wisc DeWitt	1.0	32	Datamation
1994	11.11	5.25	Alpha, Nyberg	0.2	1	Datamation
1993	1.20	0.11	Sequent, Graefe	1.0	32	Datamation
1991	14.29	0.54	IBM 3090, DFsort/Saber	2.5	1	Datamation
1990	0.31	0.15	Kitsuregawa	0.2	1	Datamation
1987	3.85	0.05	Cray YMP, Weinberger	7.0	1	Datamation
1986	0.03	0.01	Tandem Tsukerman	0.3	3	Datamation
1985	0.02	0.05	M6800 Bitton et al	0.03	1	Datamation

Table 5: Performance/Price result for SheenkSort and historical ones

Region Consideration

The sorting benchmarks are accepted more and more widely in the world. But the four web sites to purchase the hardware components are all in U.S. It is difficult for those out of U.S. to buy anything from these sites. All hardware components of SheenkSort were bought from a local market and assembled in China. The overall price is almost the same as in U.S., but we still met

[17] Gray, J., Coates J., and Nyberg C., Performance/Price Sort and PennySort. Technical Report MS-TR -98-45, Microsoft Research, August 1998. http://research.microsoft.com/barc/SortBenchmark/PennySort.doc.

[18] The original text is price per second. It is revised to price per minute because the time in item (1) is minute.

the problem that some devices were never sold in U.S. and could not be our choices. Only the devices sold both in U.S. and in China might be considered. As an international benchmark, PennySort as well as the other ones is hoped that the region problem might be considered more in the future.

Acknowledgements:

Special thanks to Mr. Jim Gray who set up the benchmarks. We also thank Peng Liu, the 2002 Daytona PennySort winner who gave us generous help.