2000 Performance / Price Sort and PennySort

Brad Helmkamp
Keith McCready

April 5th, 2000

Research and Development
Stenograph LLC
1500 Bishop Ct.
Mt. Prospect, IL 60056

2000 Performance / Price Sort and PennySort

Brad Helmkamp, Keith McCready, Stenograph LLC
helmk@bigfoot.com keithmc@ix.netcom.com
March 2000

Abstract: HMSort is a minimal functionality external sort written to demonstrate the advantages of combining overlapped IO with a well thought out sorting algorithm. The PennySort, a type of sorting benchmark designed to run on inexpensive machines available to the average user, is a good place to demonstrate what can be gained by using this programming approach. HMSort was able to sort 4.19 GB [1] (45 million records) for a penny, not quite double last year's performance. This paper documents these results and also provides performance data on the Datamation, Minute and the Performance/Price sort benchmarks.

Why Stenograph LLC? Stenograph LLC is a company dedicated to providing stenographic transcriptionists with the highest quality supplies and productivity tools. Court reporters, who make up the majority of our market, use computer based dictionaries of up to 250,000 entries to provide a mapping from their shorthand to English words and phrases. Managing these dictionaries has traditionally been a time consuming activity with limited functionality. With the use of efficient sorting, we are removing the frustrations of waiting on the software to complete dictionary tasks and providing functionality that was not available before. By entering the PennySort competition we hoped to learn from and compare ourselves to the current sort programs on the market.

2000 Indy PennySort

Our System Choice: Last year as we tested our sort program HMSort, we found that sort times were very much I/O bound. That is, at a given CPU clock speed, our sort ran faster on systems with fast hard drives and disk controllers and slower on systems with slow hard drives and/or controllers. In order to have the best performance for the price, we shopped around to find the fastest of the lower cost hard drives. We found some Maxtor DiamondMax+ drives fit the bill. Satisfied with the performance of these drives we began to purchase them for other systems. This year, we were in the process of upgrading some of our development workstations and so we purchased a couple of the latest hard drives from Maxtor, in the DiamondMax Plus 40 series. Before we put these drives into our workstations, we thought that we ought to see how HMSort performed on them.

We put the drives in a 500Mhz Pentium III system with 128MB of RAM (see Table 1) along with a Promise Ultra66 ATA controller to maximize the drive throughput. When we tested these drives with WinBench99, we found that they had a sequential transfer rate of just over 30MB/s (to start with on the outer tracks – for a in-depth evaluation, see http://www.storagereview.com). This was just about exactly twice the rate observed last year on our DiamondMax+ 6800 series drives.

Our PennySort system is described in Table 1:

Table 1: Price breakdown of 2000 PennySort Machine

Quantity	Description	Cost	Total
2	10 GB UDMA 7200 rpm Maxtor DiamondMax+ Hard Drive	$139.00	$278.00
1	500 Pentium III Processor	$319.00	$319.00
1	128 MB RAM	$146.00	$146.00
1	Windows 2000 Operating System Upgrade	$119.00	$119.00
1	3.5 inch Floppy Drive	$15.00	$15.00
1	ACER 10/100Mbps Network Card	$25.00	$25.00
1	Promise Ultra66 ATA Controller	$38.00	$38.00
1	Shipping	$48.08	$48.08
Total			$1010.00

Noting that our new drives had twice the sequential transfer rate as last year's, we speculated that our 1-pass Datamation sort just might take half the time it did last year. We further speculated that 2-pass sorts would not show a similar 2x improvement as the same amount of seeking on the drives would be required and the average seek time for the newer drives is the same as it was last year. So the more seeking required, the more the results would resemble last year's results. The results below confirm our speculation.

The time budget for the PennySort assumes a three year depreciation of the hardware and is calculated, therefore, by dividing the number of seconds in three years (94,608,000) by the system cost. Our system cost of $1010.00 yields 936 seconds/penny.

Results: We generated files with 100-byte records using a modified[2] version of the SortGen program.[3] We ran each of the sorts with several different sizes of files searching for the maximum amount they could sort in 936 seconds.

The results are recorded in table 2.

Table 2: 2000 PennySort Times.

Product	Time Budget	Best Time	Kernel	User	Total CPU Time	Sorted GB	Category
HMSort	936	886	25.8	445	471	4.19	Indy

Datamation Sort Results

The Datamation sort benchmark tests a one pass sort of one million 100 byte records. On our system this is a one pass sort, so reading and writing cannot be overlapped with each other. The Datamation sort, therefore, places more emphasis on the efficiency of the sorting algorithm used and on how much of its computation can be overlapped with either reading or writing. Normally the Datamation sort does not take into account the cost of the system (the 2000 winner finished in 0.998 second at a cost of $80,000), but because it shows the speed of a one-pass sort it is valuable for comparing how the PennySort entrants perform on smaller files.

HMSort was able to sort 1 million records in 8 seconds. This is exactly twice as fast as the result we achieved last year. We attribute this to having hard drives with twice the throughput of last year's drives. We were a little worried that our sort algorithm might start to become processor bound rather than I/O bound, but this proved not to be the case. Wanting to explore this further, we modified our Ultra66 controller to act as a RAID controller. We benchmarked the two drives together as a stripe set at 50MB/s. We hoped to see a Datamation sort result around 4.8 seconds. Unfortunately, the modification we performed on the controller did not prove to be stable and the controller reverted to operating at a lower speed.

Minute Sort Results

The Minute sort benchmark is simply "how many 100 byte records can you sort in one minute." Because on our system this is a two-pass sort, reading and writing are overlapped with each other. The Minute sort, therefore, places more emphasis on efficiently overlapping I/O and the raw speed of the disks used. The Minute sort result is used below in the Performance/Price calculation of GB/$.
HMSort was able to sort 3.9 million records in 60 seconds.

2000 Performance/Price Sort

(1) Sort the largest file you can in a minute.
(2) Divide file size in GB by 60 to calculate GB/s.
(3) Compute the system price in $ per second (3-year depreciation => system price divided by 9.5e-7).
(4) Compute the GB/$ sorted by dividing item 2 by item 3.

The Performance/Price results as recorded in last year's PennySort document[4] are shown in Table 3 with our 2000 results added onto the end:

Table 3: Historical Performance/Price results.

year	MB/sec	GB/$	System	Sys price (M$)	CPUs	
1985	0.02	0.05	M6800 Bitton et al	0.03	1	Datamation
1986	0.03	0.01	Tandem Tsukerman	0.3	3	Datamation
1987	3.85	0.05	Cray YMP, Weinberger	7.0	1	Datamation
1991	14.29	0.54	IBM 3090, DFsort/Saber	2.5	1	Datamation
1990	0.31	0.15	Kitsuregawa	0.2	1	Datamation
1993	1.20	0.11	Sequent, Graefe	1.0	32	Datamation
1994	1.72	0.16	IPSC/Wisc DeWitt	1.0	32	Datamation
1994	11.11	5.25	Alpha, Nyberg	0.2	1	Datamation
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
1996	100.00	15.76	NOW, Arpaci-Dusseau	0.6	32	Minute/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
1998	1.74	125.00	PostmanSort	0.0013	1	Penny/Daytona
1998	1.74	144.00	NTSort	0.0012	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
2000	**6.50**	**608.86**	**HMSort**	**0.0010**	**1**	**Penny/Indy**

[1] Throughout this paper KB=1024 bytes, MB=1024[2] bytes, and GB=1024[3] bytes. Therefore 45,000,000 records of 100 bytes each = 4.19 GB
[2] We modified SortGen.c to open its file in binary mode instead of text mode. This prevents fwrite from translating the "\x0d\x0a" into "\x0d\x0d\x0a" so the records are exactly 100-bytes as specified in the benchmark rules.
[3] SortGen at http://research.microsoft.com/barc/SortBenchmark
[4] PennySort.doc at http://research.microsoft.com/barc/SortBenchmark