The description of program for Pennysort

Xing Huang, Bo Huang

(Coach: Binheng Song)

huangx04@mails.tsinghua.edu.cn

lavahx@hotmail.com

March 30, 2006

Abstract: Bytes-split index Sort: bytes-split index sort (BSIS), is an non-compare-based sorting algorithm. The algorithm splits every digit of a sequence into m (m>= 2) bytes and constructs m subsequences, then sorts every subsequence with an index technique similar to self-indexed sort [1]. This results on time complexity O(n) on uniform data. BSIS can sort 344 million records for a Penny using April 2006 PC pricing.

School of Software,

Tsinghua University, Beijing 100084, China

Department of Mathematics & Computer Science

Hunan Industry University, Zhuzhou 421007,

1. Internal Sorting algorithm

Bytes-split index sort (BSIS), on an approach of non compare-based sorting. The algorithm splits every digit of a sequence into m (m>= 2)bytes and constructs m subsequences, then sorts every subsequence via using index technique which is similar to self-indexed sort[1]. Results on time complexity O(n) is achieved..

1.1 Example of Sorting int32

Source data: `89520564, 9851258, 11258472`

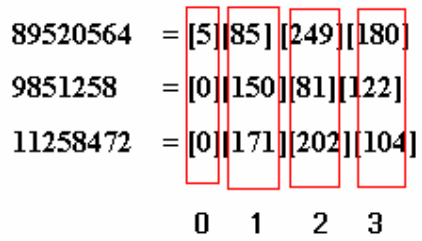
1.1.1 Splitting source data

We can use a union type to split source data.

```
typedef union
{
    int         Data ;
    byte        bdata4[4];
} S_DATA4;
Original sequence:
89520564  = [5][85] [249][180]
9851258   = [0][150][81][122]
11258472  = [0][171][202][104]
```

1.1.2 Sorting subsequences

Now there are four subsequences



We can use a way to sort every subsequences. The way is similar to SIS[1].

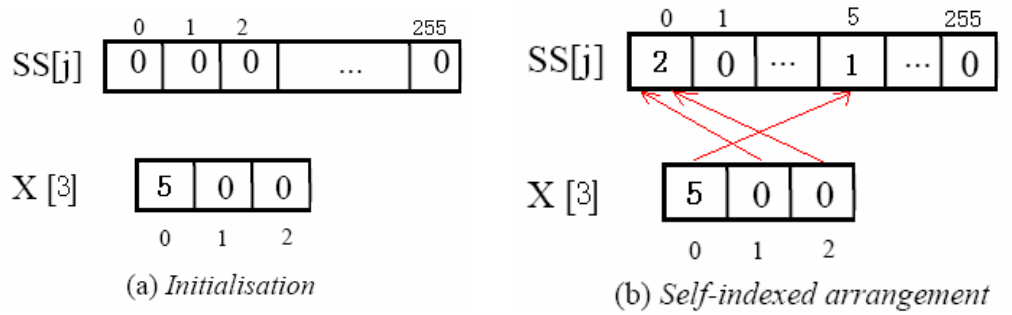
A unsigned byte max value is 256.

SS is a 256-byte buffer for sorting.

X is the first subsequence.

DataBufer is same size as source data .

The following is the process of sorting one subsequence.



Because X is just a part of souce data, the result of SS can not be directly copied to X which would destroy the source data. So, the SS result is saved to a DataBufer. Then the result of DataBufer replaces the source data. After first sorting, we get the sequence.

```
9851258   = [0][150][81][122]

11258472  = [0][171][202][104]

89520564  = [5][85] [249][180]
```

BSIS can recursively handle other subsequences.

1.2 Example of Sorting 100-byte string

source data:

"!@>{_+4ewf AAAAAAAAAAJJJJJJJJJ....."

"~!>V(%$8 BBBBBBBBBBHHHHHHHHHH....."

We can use the same way to sort 100 subsequences. The way is similar to SIS[1].

Max value of the string is 127.

SS is a 127-byte buffer for sorting.

X is the first subsequence.

DataBufer is same size as source data .

The following process is same as Example of Sorting int32.

The speed of sorting 100MB 100-byte string is 1.48 seconds on a pc.

To optimizie sort, we can convert every 2 characters to an unsigned short int which speeds up sorting.

1.3 Performance in average case

The performances of BSIS and qucik sort algorithms are comparaed under average cases by using random test data from size 100000 to 20000000. The result is given in Table1-1.

Table 1-1 (Seconds to sort N bytes)

Size (N)	BSIS(m=2)	BSIS(m=4)	QuickSort
100,000	0.500	0.016	0.031
500,000	0.500	0.047	0.140
1,000,000	0.547	0.094	0.344
5,000,000	0.750	0.531	5.360
10,000,000	1.016	1.062	19.750
20,000,000	1.579	2.141	81.922

2. Process of the program for pennysort

Our program fell into two successive phases.

2.1 First phase

First phases is splitting source data file to almost 100 small files.

2.1.1 Reading source data file

Under windows xp there are four ways to read source data: sequential read, memory map, overlapped I/O and complete port.

Table 2-1 Speed (Bytes/Second) of four read methods

	sequential read	memory map	overlapped I/O	IO completion port
speed(MB/S)	86	95	114	147

So we choice the IO completion port approach to read source data.

2.1.2 Splitting source data to small files

We regard first char of every 100-byte string as a key to split source data. That means all 100-byte string will be in a same small file if their first char is same. The speed of first phase is ***65.69MB/S.***

2.2 Second phase

Second phase is reading every file and handling them. We just sequentially read, sort and write data. The speed of second phase is ***27.82MB/S.***

3. How to get our program

I can provide two programs: one for pennysort in several days, the other is BSIS for general purpose in 2 weeks.

I can send them to you or others can send an email tome and I send programs to them.

4. Hardware/OS Configuration:

Table 1: Price list of the Machine

Quantity	Description		Cost	Total
1	ASUS A8N-VM CSM Socket 939 RETAIL[1]	NVIDIA GeForce 6150	$78.99	$78.99
		10/100/100MB Marvell 88E1111 PHY		
		Audio ADI AD 1986A		
4	HITACHI 7K80 80GB SATAII 7200RPM [2]		$50.00	$200.00
2	APACER PC3200 512MB DDR400[3]		$31.50	$62.00
1	AMD ATHLON64 3000 + 936 1.8G RETAIL BOX CPU[4]		$145.00	$145.00
1	CPU Fan[5]		7.99$	7.99$
1	ATX Case, 300W, UL, Middle Tower		$43.50	$43.50
1	Assembly		$25.00	$25.00
1	Microsoft Windows XP Professional Upgrade with SP2 - Retail [6]		NA	NA
Total price	**$563**			
Time budget	94608000/56300 = 1,680 seconds			

Results:

Sort Algorithm	Time Budget	Best Time	Sorted bytes	Category
BSISort	1,680 sec	1,679 sec	34,406,000,000	Daytona

5. Results

Sort Algorithm	Time Budget	Best Time	Sorted bytes	Category
BSISort	1,680 sec	1,679 sec	34,406,000,000	Daytona

Reference

[1]Yingxu Wang.A New Sort Algorithm: Self-Indexed Sort.Communications of ACM SIGPALN, 1996,Vol.31, No.3, March:28-36