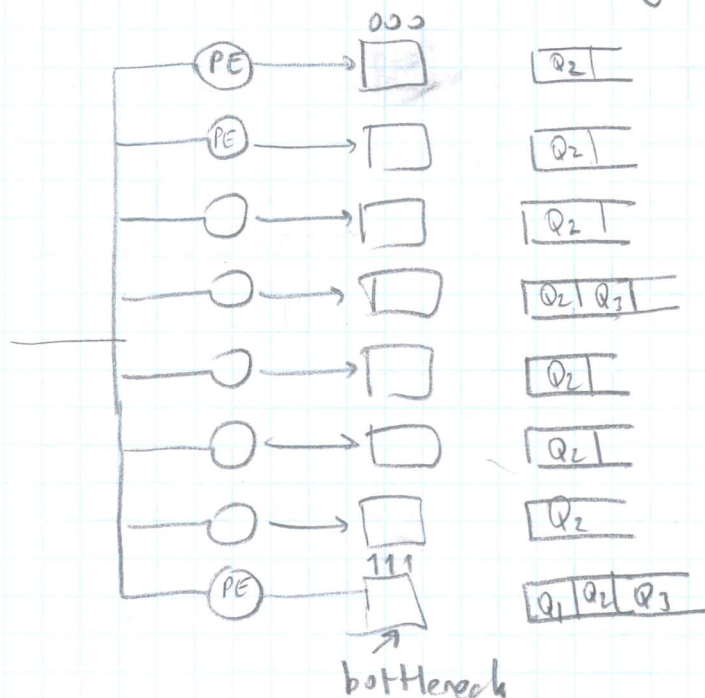


Allocate more PE's to the partitions with more # of 1's.

PE: Processing elements

### Correction Machine



### Sequential Processing

$$Q_1 = 1 \text{ time unit}$$

$$Q_2 = 8 \text{ tu}$$

$$Q_3 = 2 \text{ tu}$$

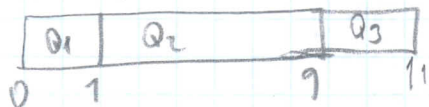
### Parallel Processing

$$Q_1 = 1 \text{ tu}$$

$$Q_2 = 2 \text{ tu}$$

$$Q_3 = 3 \text{ tu}$$

### Processing time



Arrival time for all queries = 0 tu

$$\text{Serial turnaround time} = [(1-0) + (9-0) + (11-0)] / 3 = 24/3 = 8 \text{ tu}$$

$$\text{Parallel turnaround time} = [(1-0) + (2-0) + (3-0)] / 3 = 6/3 = 2 \text{ tu}$$

Serial Throughput := # of queries completed / total time: 3/11

Parallel throughput := 3/3 = 1

Speedup Ratio:  $\frac{\text{Total processing time in serial}}{\text{Total processing time in parallel}}$

## Linear Hashing with Superimposed Signature

R. Zezula, Tiberio, Ciaccia

ACM TOIS 1991

Quick filter

S1	0011	1000
S2	0100	0011
S3	0010	1011
S4	0100	1111
S5	0000	1111
S6	0001	1011
S7	0001	0011
S8	0100	0000
S9	0001	1100
S10	1100	0100
S11	1101	0100
S12	1111	0010

$bv = 0$

$Bkfr = 3$

$Lf = 2/3$

### Insertion Algorithm

Use the hsuffix to find the page to be inserted.

Update File When LF of the file

is at the desired level after

adding  $Lf \times Bkfr$  no. of signatures

add one more block to the file and

distribute the records of the bv block between this new block and the bv block.

### Updating Bv

$bv = bv + 1$

if  $bv = 2^h$  then

$h = h + 1$

$bv = 0$

$bv = 0$

S1		
S2	S3	S4

$Lf = \frac{4}{2 \times 3} = \frac{2}{3}$

$br = 0$

S1		
S2	S3	S4

→

S5	S6
----	----

$br = 1$

S1		
S2	S3	S4

→

S7	S8
----	----

00	s1	s8	
1	s2	s3	s4
10			

→ 

s5	s8	s7
----	----	----

Insert s8

00	s1	s8	
01			
10			
11	s2	s3	s4

→ 

s5	s6	s7
----	----	----

Insert s9, s10

00	s1	s8	s9
01			
10			
11	s2	s3	s4

→ 

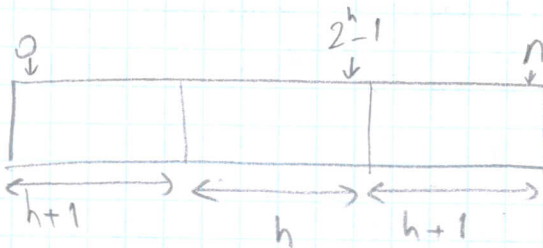
s5	s6	s7
----	----	----

Update file :

000	s1	s8	
01			
10			
11	s2	s3	s4
100	s9	s10	

→ 

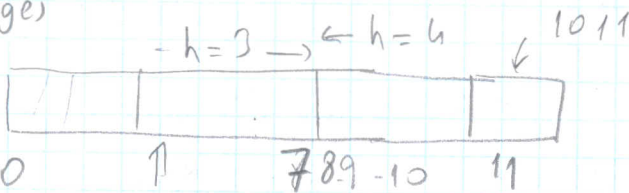
s5	s6	s7
----	----	----



$[bv, 2^h - 1]$  at level  $h$

$n=9 \Rightarrow$  hashing level 3

12 pages



12 pages ma xpage  $\approx .11 = 1011$

0, 1, 2, 3



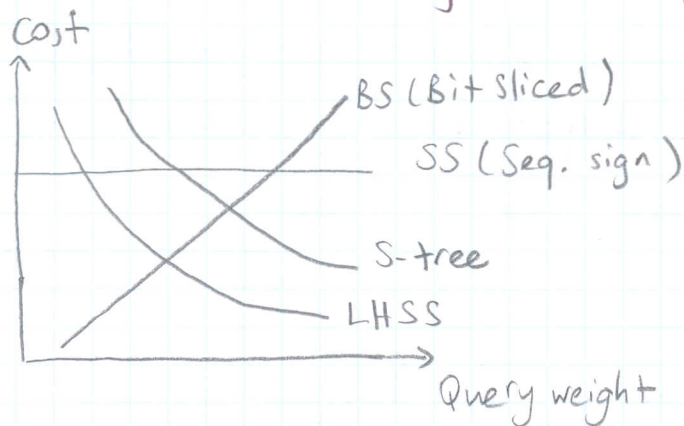
$$Q_1 = 0010 \ 0110$$

Use  $h$  or  $(h+1)$  suffix of the query according to the page hashing level.

if  $((Q_k \text{ and } P_k) = Q_k)$  then  
access the page

Page Key	Query Key	Page Key & Query
000	110	000
001	110	000
10	10	10
11	10	10
100	110	100
101	110	100

LHSS: Linear Hashing with Superimposed Formula



Performance Evaluation of LHSS

$W(Q)$ : query weight (no. of 1's)

$EXPH(W(Q), h)$ : Expected no. of bits set in the  $h$ -suffix of the query signature

$F$ : signature size in bits

$$EXPH(W(Q), h) = \sum_{j=1}^{\min(h, W(Q))} j \times P(j)$$

$P(j)$ : probability that  $j$  bits are set in the  $h$ -suffix of the query.

$$P(j) = \frac{\binom{h}{j} \binom{F-h}{W(Q)-j}}{\binom{F}{W(Q)}}$$

pick  $j$  out of  $h$  ← pick the rest of the query bits from the remaining part of the query. → different ways of picking  $W(Q)$  bits from  $F$

Probability of Access Savings =  $P(w(Q), h)$

= proportion of the pages that do not need to be accessed.

$$P(w(Q), h) = 1 - \frac{npa}{n}$$

$npa$  = no. of pages accessed

$n$  = no of pages in signature file

No. of pages to be accessed =  $2^{h - \text{EXPH}(w(Q), h)}$

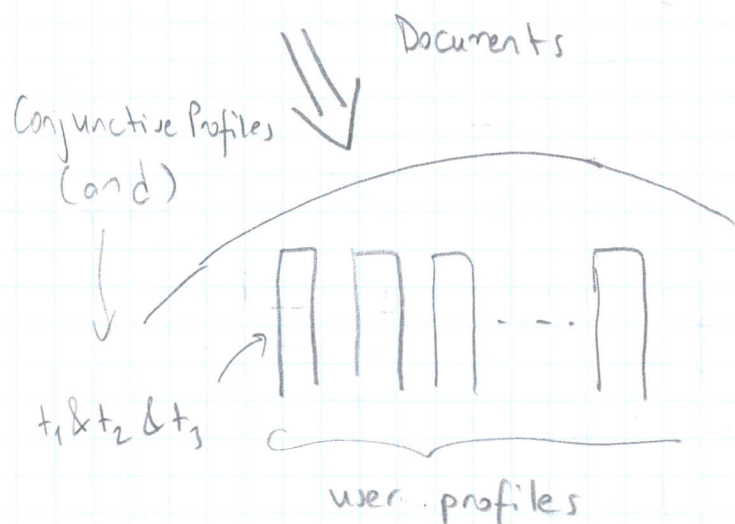
When  $n = 2^h$

$$npa = \frac{2^h}{2^{\text{EXPH}(w(Q), h)}}$$

$$P(w(Q), h) = 1 - \frac{n / 2^{\text{EXPH}(w(Q), h)}}{n} = 1 - \frac{1}{2^{\text{EXPH}(w(Q), h)}}$$

## Information Filtering

### Selective Dissemination of Information



push-system

IR: pull documents

IR: IF

"IR & IF": Two sides of the same coin, Communications of the ACM, Craft and Belkin

ACM TODS - Yan, T.W., Garcia-Molina, H.

Index structure for Selective Dissemination of Information under the Boolean Model, 1994