CS 533 Lecture Notes
12.05.2014
Devrim Şahin

Signature File Partitioning (cont.)

- Fixed prefix

Use the initial part as a key

We want to consider both sequential and parallel processing

Partition activation ratio (PAR), signature activation ratio (SAR) (covered last time)

S1: 0111 1000 k=2
S2: 1000 1011 (Use 2-prefix)
S3: 0011 1100
S4: 1100 0011
S5: 0110 1100
S6: 1001 0011
S7: 0000 1111

00	01	10	11
S3 S7	S1 S5	S2 S6	S4

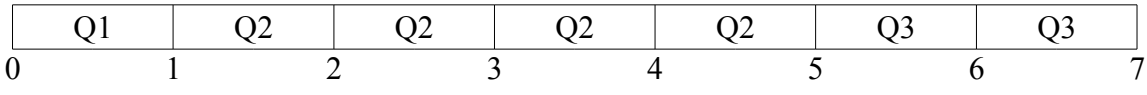
Q1: 1110 0001
Q2: 0000 1111
Q3: 0110 0011

We select the pages of which the prefixes P_i satisfy the rule P_i & $Q_i = Q_i$.

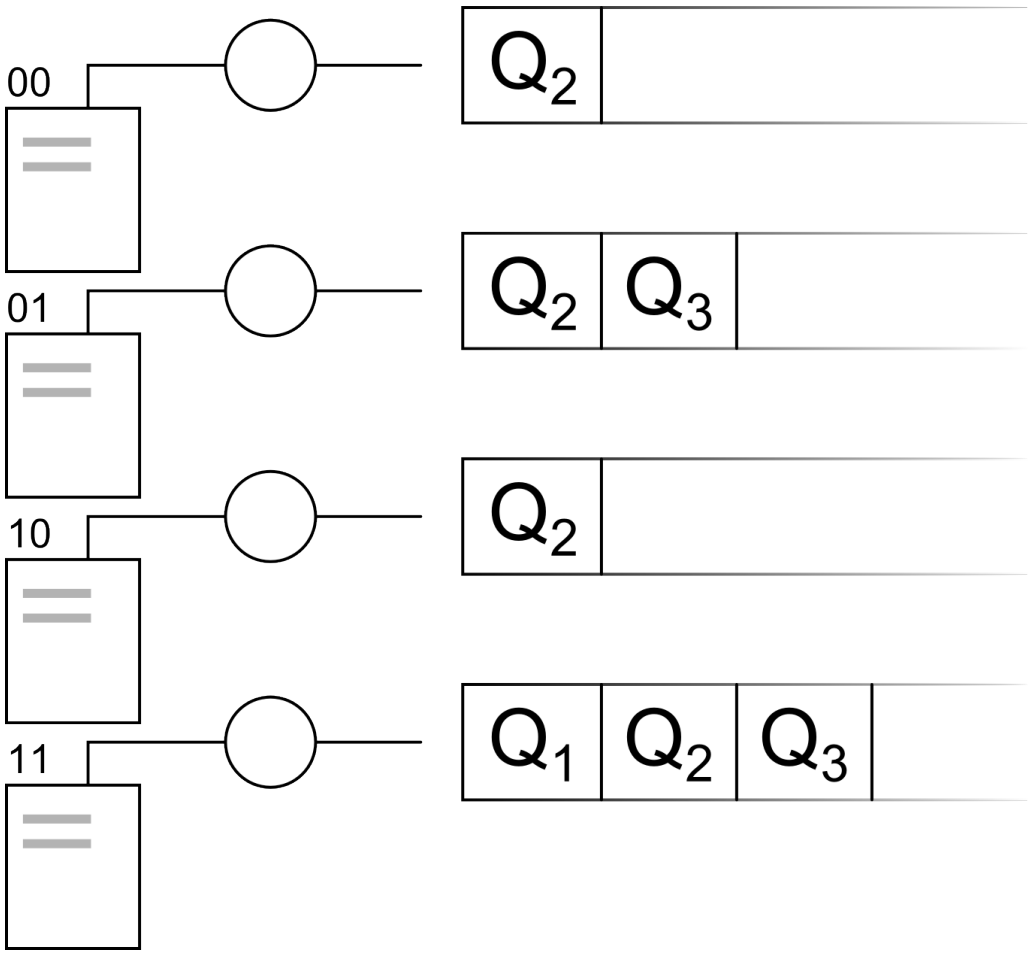
	k = 2
Q1: 111...	11 (**1 page selected)**
Q2: 000...	00, 01, 10, 11 **(4)**
Q3: 011...	01, 11 **(2)**

Sequential processing

All queries arrive at t=0



Parallel processing



Turnaround time = Time of completion – Time of arrival

	Time of arrival	
	Seq.	**Par.**
Q1	1	1
Q2	5	2
Q3	7	3

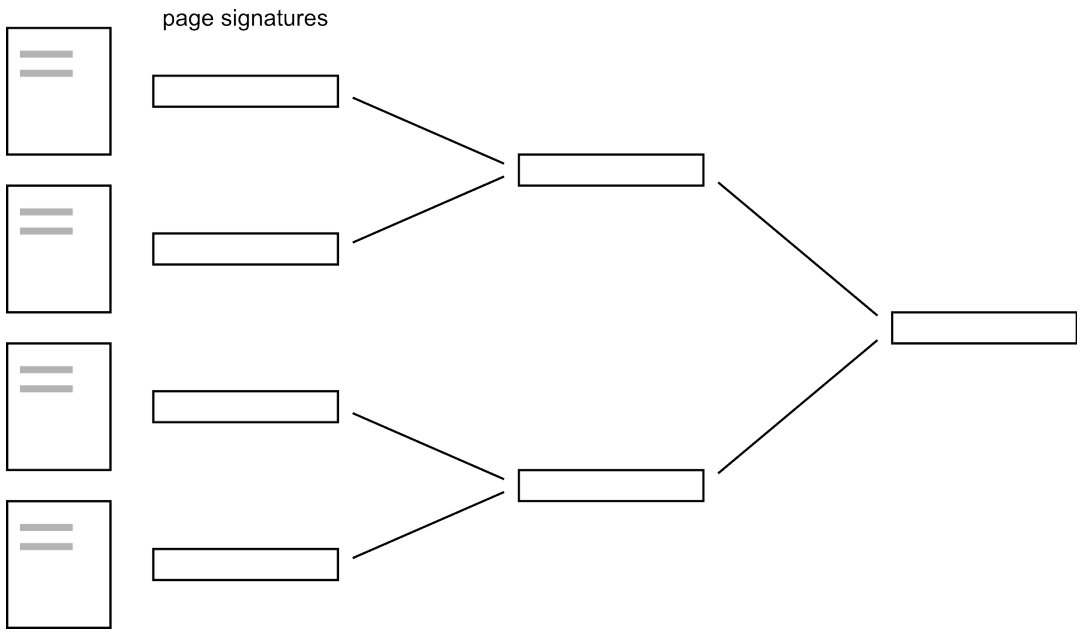
Serial average turnaround time = (1+5+7) = 3 tu (tu: Time units)

Parallel average turnaround time = (1+2+3) = 2 tu

Throughput (T): No. of jobs completed per unit time

$T_s = 3/7$ $T_p = 3/3$ $T_p > T_s$

Signature Tree Structure



Page signatures are ANDed (“superposed”) to obtain superpage signatures. Any branch that do not satisfy the query condition is simply pruned.

Information Filtering

- Also known as “selective dissemination of information”
- It is the mirror image of information retrieval
- Change the roles of queries and documents

User profiles ≡ Queries

We receive **documents**, and we match these **documents** with **user profiles**. That is, we send the incoming document to the matching user profiles. Therefore the process is reversed.

We can convert the documents to vectors, and calculate similarity

Or, if an incoming document contains all user profile terms, send it to the owner of the profile (we use the AND operand).

(see: Yan, Tak W., and Héctor García-Molina. "Index structures for selective dissemination of information under the boolean model." *ACM Transactions on Database Systems (TODS)* 19.2 (1994): 332-364.)

There are several approaches described in the paper:

Brute force approach (Sequential comparison):

We compare the incoming document with each user profile one by one

Counting method:

Sample profiles:

p_1: (a, b)

p_2: (a, d)

p_3: (a, d, e)

p_4: (b, f)

p_5: (c, d, e, f)

Sample document: a, c, a, f, b, c

Unique terms: {a, b, c, f}

Construct an inverted index:

Directory (in memory)		Inverted list (on disk)		
a	→	p_1	p_2	p_3
b	→	p_1	p_4	
c	→	p_5		
d	→	p_2	p_3	p_5
e	→	p_3	p_5	
f	→	p_4	p_5	

Before processing a document set all count entries equal to 0.

Take the unique terms of the incoming document, visit the posting list and increment the counts for matching profiles.

	Total
p_1	2
p_2	2
p_3	3
p_4	2
p_5	4

	a	b	c	f
p_1	0	1	2		
p_2	0	1			
p_3	0	1			
p_4	0		1		2
p_5	0			1	2

In this example, we obtained the required counts for p_1 and p_4; so they will receive the document.

Key method

Assume that we know the occurence frequency of the terms in documents:

a b c d e f

↑ ↑

Least frequent term Most frequent term

A profile appears in one of the inverted lists only in its term that appears most infrequently in documents:

Directory Inverted list (posting list)

a	→	p_1	1	b	p_2	1	d	p_3	2	d	e
b	→	p_4	1	f						
c	→	p_5	3	d	e	f				
d	→									
e	→									
f	→									

We skip these as stopwords

The expectation: The term that appears less frequently in documents will appear more frequently in user profiles.

Exercises:

1. Distribute the signatures below into pages with (a) k=2 and (b) k=3 prefix.

S1 = 0011 1001, S2 = 1010 1100, S3 = 0101 0101, S4 = 1111 0000,
S5 = 0001 1110, S6 = 1101 0100, S7 = 1001 0011, S8 = 0110 0110

a. For k=2, pages have these prefixes: P0=00, P1=01, P2=10, P3=11.

Then P0 = {S1, S5}; P1 = {S3, S8}, P2 = {S2, S7}, P3 = {S4, S6}

b. For k=2, pages are P0=000, P1=001, P2=010, P3=011, P4=100, P5=101, P6=110, P7=111.

Then P0 = {S5}, P1={S1}, P2={S3}, P3={S8}, P4={S7}, P5={S2}, P6={S6}, P7={S4}.

2. For both cases, which page gets retrieved the most? Why?

For k=2, P3 gets retrieved every time. For k=3, P7 gets retrieved every time. The reason is that these prefixes are all 1s, and P_i & Q_i = Q_i is satisfied for any query.