Sequential: Vector of Bits

When estimating seek costs, we need to calculate the probability distribution for the distance between two subsequent qualifying cylinders. We model the situation as a bitvector of length B with b bits set to one. Then, B corresponds to the number of cylinders and a one indicates that a cylinder qualifies.
[Later: Vector of Buckets]

Sequential: Vector of Bits (2)

Theorem

Assume a bitvector of length B. Within it b ones are uniformly distributed. The remaining $B - b$ bits are zero. Then, the probability distribution of the number j of zeros

1. *between two consecutive ones,*
2. *before the first one, and*
3. *after the last one*

is given by

$$\mathcal{B}_b^B(j) = \frac{\binom{B-j-1}{b-1}}{\binom{B}{b}} \tag{35}$$

Sequential: Vector of Bits (3)

Proof:
To see why the formula holds, consider the total number of bitvectors having a one in position i followed by j zeros followed by a one.
This number is

$$\binom{B-j-2}{b-2}$$

We can chose $B-j-1$ positions for i.
The total number of bitvectors is

$$\binom{B}{b}$$

and each bitvector has $b-1$ sequences of the form that a one is followed by a sequence of zeros is followed by a one.

Sequential: Vector of Bits (4)

Hence,

$$\begin{aligned} \mathcal{B}_b^B(j) &= \frac{(B-j-1)\binom{B-j-2}{b-2}}{(b-1)\binom{B}{b}} \\ &= \frac{\binom{B-j-1}{b-1}}{\binom{B}{b}} \end{aligned}$$

Part (1) follows.
To prove (2), we count the number of bitvectors that start with j zeros before the first one.
There are $B-j-1$ positions left for the remaining $b-1$ ones.
Hence, the number of these bitvectors is $\binom{B-j-1}{b-1}$ and part (2) follows.
Part (3) follows by symmetry.

Sequential: Vector of Bits (5)

We can derive a less expensive way to calculate formula for $\mathcal{B}_b^B(j)$ as follows.
For $j = 0$, we have $\mathcal{B}_b^B(0) = \frac{b}{B}$.
If $j > 0$, then

$$\begin{aligned}
\mathcal{B}_b^B(j) &= \frac{\binom{B-j-1}{b-1}}{\binom{B}{b}} \\
&= \frac{\frac{(B-j-1)!}{(b-1)!((B-j-1)-(b-1))!}}{\frac{B!}{b!(B-b)!}} \\
&= \frac{(B-j-1)!\ \ b!(B-b)!}{(b-1)!((B-j-1)-(b-1))!\ \ B!}
\end{aligned}$$

Sequential: Vector of Bits (6)

$$
\begin{aligned}
\mathcal{B}_b^B(j) &= \frac{(B-j-1)!\ \ b!(B-b)!}{(b-1)!((B-j-1)-(b-1))!\ \ B!} \\
&= b\frac{(B-j-1)!\ \ (B-b)!}{((B-j-1)-(b-1))!\ \ B!} \\
&= b\frac{(B-j-1)!\ \ (B-b)!}{(B-j-b)!\ \ B!} \\
&= \frac{b}{B-j}\frac{(B-j)!\ \ (B-b)!}{(B-b-j)!\ \ B!} \\
&= \frac{b}{B-j}\prod_{i=0}^{j-1}(1-\frac{b}{B-i})
\end{aligned}
$$

This formula is useful when $\mathcal{B}_b^B(j)$ occurs in sums over j.

Sequential: Vector of Bits (7)

Corollary

Using the terminology of Theorem 8, the expected value for the number of zeros

1. *before the first one,*
2. *between two successive ones, and*
3. *after the last one*

is

$$\overline{\mathcal{B}}_b^B = \sum_{j=0}^{B-b} j\mathcal{B}_b^B(j) = \frac{B-b}{b+1} \tag{36}$$

Sequential: Vector of Bits (8)

Proof:

$$\begin{aligned}
\sum_{j=0}^{B-b} j\binom{B-j-1}{b-1} &= \sum_{j=0}^{B-b}(B-(B-j))\binom{B-j-1}{b-1} \\
&= B\sum_{j=0}^{B-b}\binom{B-j-1}{b-1} - \sum_{j=0}^{B-b}(B-j)\binom{B-j-1}{b-1} \\
&= B\sum_{j=0}^{B-b}\binom{b-1+j}{b-1} - b\sum_{j=0}^{B-b}\binom{B-j}{b} \\
&= B\sum_{j=0}^{B-b}\binom{b-1+j}{j} - b\sum_{j=0}^{B-b}\binom{b+j}{b}
\end{aligned}$$

Sequential: Vector of Bits (9)

$$\begin{aligned}
\sum_{j=0}^{B-b} j\binom{B-j-1}{b-1} &= B\sum_{j=0}^{B-b}\binom{b-1+j}{j} - b\sum_{j=0}^{B-b}\binom{b+j}{b} \\
&= B\binom{(b-1)+(B-b)+1}{(b-1)+1} - b\binom{b+(B-b)+1}{b+1} \\
&= B\binom{B}{b} - b\binom{B+1}{b+1} \\
&= (B - b\frac{B+1}{b+1})\binom{B}{b}
\end{aligned}$$

With

$$\begin{aligned}
B - b\frac{B+1}{b+1} &= \frac{B(b+1)-(Bb+b)}{b+1} \\
&= \frac{B-b}{b+1}
\end{aligned}$$

the claim follows

Sequential: Vector of Bits (10)

Corollary

Using the terminology of Theorem 8, the expected total number of bits from the first bit to the last one, both included, is

$$\overline{\mathcal{B}}_{tot}(B, b) = \frac{Bb + b}{b + 1} \tag{37}$$

Sequential: Vector of Bits (11)

Proof:
We subtract from B the average expected number of zeros between the last one and the last bit:

$$\begin{aligned} B - \frac{B-b}{b+1} &= \frac{B(b+1)}{b+1} - \frac{B-b}{b+1} \\ &= \frac{Bb + B - B + b}{b+1} \\ &= \frac{Bb+b}{b+1} \end{aligned}$$

Sequential: Vector of Bits (12)

Corollary

Using the terminology of Theorem 8, the number of bits from the first one and the last one, both included, is

$$\overline{\mathcal{B}}_{1\text{-span}}(B, b) = \frac{Bb - B + 2b}{b + 1} \tag{38}$$

Sequential: Vector of Bits (13)

Proof (alternative 1):
Subtract from B the number of zeros at the beginning and the end:

$$\begin{aligned}\overline{\mathcal{B}}_{\text{1-span}}(B,b) &= B - 2\frac{B-b}{b+1} \\ &= \frac{Bb + B - 2B + 2b}{b+1} \\ &= \frac{Bb - B + 2b}{b+1}\end{aligned}$$

Sequential: Vector of Bits (14)

Proof (alternative 2):
Add the number of zeros between the first and the last one and the number of ones:

$$\begin{aligned}
\overline{\mathcal{B}}_{\text{1-span}}(B,b) &= (b-1)\overline{\mathcal{B}}_b^B + b \\
&= (b-1)\frac{B-b}{b+1} + \frac{b(b+1}{b+1} \\
&= \frac{Bb - b^2 - B + b + b^2 + b}{b+1} \\
&= \frac{Bb - B + 2b}{b+1}
\end{aligned}$$

Sequential: Applications for Bitvector Model

- If we look up one record in an array of B records and we search sequentially, how many array entries do we have to examine on average if the search is successful?
- Let a file consist of B consecutive cylinders. We search for k different keys all of which occur in the file. These k keys are distributed over b different cylinders. Of course, we can stop as soon as we have found the last key. What is the expected total distance the disk head has to travel if it is placed on the first cylinder of the file at the beginning of the search?
- Assume we have an array consisting of B different entries. We sequentially go through all entries of the array until we have found all the records for b different keys. We assume that the B entries in the array and the b keys are sorted. Further all b keys occur in the array. On the average, how many comparisons do we need to find all keys?

Sequential: Vector of Buckets

Theorem (Yao)

Consider a sequence of m buckets. For $1 \leq i \leq m$, let n_i be the number of items in a bucket i. Then there is a total of $N = \sum_{i=1}^{m} n_i$ items. Let $t_i = \sum_{l=0}^{i} n_i$ be the number of items in the first i buckets. If the buckets are searched sequentially, then the probability that j buckets that have to be examined until k distinct items have been found is

$$C_{n_i}^{N,m}(k,j) = \frac{\binom{t_j}{k} - \binom{t_{j-1}}{k}}{\binom{N}{k}} \tag{39}$$

Thus, the expected number of buckets that need to be examined in order to retrieve k distinct items is

$$\overline{C}_{n_i}^{N,m}(k) = \sum_{j=1}^{m} j C_{n_i}^{N,m}(k,j) = m - \frac{\sum_{j=1}^{m} \binom{t_{j-1}}{k}}{\binom{N}{k}} \tag{40}$$

Sequential: Vector of Buckets (2)

The following theorem is very useful for deriving estimates for average sequential accesses under different models [Especially: the above theorem follows].

Theorem (Lang/Driscoll/Jou)

Consider a sequence of N items. For a batched search of k items, the expected number of accessed items is

$$A(N,k) = N - \sum_{i=1}^{N-1} Prob[Y \leq i] \tag{41}$$

where Y is a random variable for the last item in the sequence that occurs among the k items searched.

Disk Drive Costs for N Uniform Accesses

The goal of this section is to derive estimates for the costs (time) for retrieving N cache-missed sectors of a segment S from disk.
We assume that the N sectors are read in their physical order on disk.
This can be enforced by the DBMS, by the operating system's disk scheduling policy (SCAN policy), or by the disk drive controler.

Disk Drive Costs for N Uniform Accesses (2)

Remembering the description of disk drives, the total costs can be described as

$$C_{disk} = C_{cmd} + C_{seek} + C_{settle} + C_{rot} + C_{headswitch} \tag{42}$$

For brevity, we omitted the parameter N and the parameters describing the segment and the disk drive on which the segment resides.
Subsequently, we devote a (sometimes tiny) section to each summand.
Before that, we have to calculate the number of qualifying cylinders, tracks, and sectors.
These numbers will be used later on.

Number of Qualifying Cylinder

- N sectors are to be retrieved.
- We want to find the number of cylinders qualifying in extent i.
- S_{sec} denotes the total number of sectors our segment contains.
- Assume: The N sectors we want to retrieve are uniformly distributed among the S_{sec} sectors of the segment.
- $S_{\text{cpe}}(i) = L_i - F_i + 1$ denotes the number of cylinders of extent i.

Disk Costs: Number of Qualifying Cylinder

The number of qualifying cylinders in exent i is:

$S_{cpe}(i)$ * (1 - Prob(a cylinder does not qualify))

The probability that a cylinder does not qualify can be computed by deviding the total number of possibilities to chose the N sectors from sectors outside the cylinder by the total number of possibilities to chose N sectors from all S_{sec} sectors of the segment.
Hence, the number of qualifying cylinders in the considered extent is:

$$Q_c(i) = S_{\text{cpe}}(i)\mathcal{Y}^{S_{\text{sec}}}_{D_{\text{zspc}}(i)}(N) = S_{\text{cpe}}(i)(1 - \frac{\binom{S_{\text{sec}} - D_{\text{zspc}}(i)}{N}}{\binom{S_{\text{sec}}}{N}}) \qquad (43)$$

Number of Qualifying Tracks

Let us also calculate the number of qualifying tracks in a partion i.
It can be calculated by

$$S_{\text{cpe}}(i)D_{\text{tpc}}(1 - \text{Prob}(\texttt{a track does not qualify}))$$

The probability that a track does not qualify can be computed by dividing the number of ways to pick N sectors from sectors not belonging to a track divided by the number of possible ways to pick N sectors from all sectors:

$$Q_t(i) = S_{\text{cpe}}(i)D_{\text{tpc}}\mathcal{Y}^{S_{\text{sec}}}_{D_{\text{zspt}}(i)}(N) = S_{\text{cpe}}(i)D_{\text{tpc}}(1 - \frac{\binom{S_{\text{sec}} - D_{\text{zspt}}(i)}{N}}{\binom{S_{\text{sec}}}{N}}) \quad (44)$$

Number of Qualifying Tracks (2)

Just for fun, we calculate the number of qualifying sectors of an extent in zone i. It can be approximated by

$$Q_s(i) = S_{\text{cpe}}(i) D_{\text{zspc}}(i) \frac{N}{S_{\text{sec}}} \tag{45}$$

Since all $S_{\text{cpe}}(i)$ cylinders are in the same zone, they have the same number of sectors per track and we could also use Waters/Yao to approximate the number of qualifying cylinders by

$$Q_c(i) = \overline{\mathcal{Y}}^{S_{\text{cpe}}(i) D_{\text{zspc}}(S_{\text{zone}}(i)), S_{\text{cpe}}(i)}_{D_{\text{zspc}}(S_{\text{zone}}(i))}(Q_s(i)) \tag{46}$$

If $Q_s(i)$ is not too small (e.g. > 4).

Command Costs

The command costs C_{cmd} are easy to compute. Every read of a sector requires the execution of a command. Hence

$$C_{cmd} = ND_{\text{cmd}}$$

estimates the total command costs.

Seek Costs

- often the dominant part of the costs
- we look at several alternatives from less to more precise models

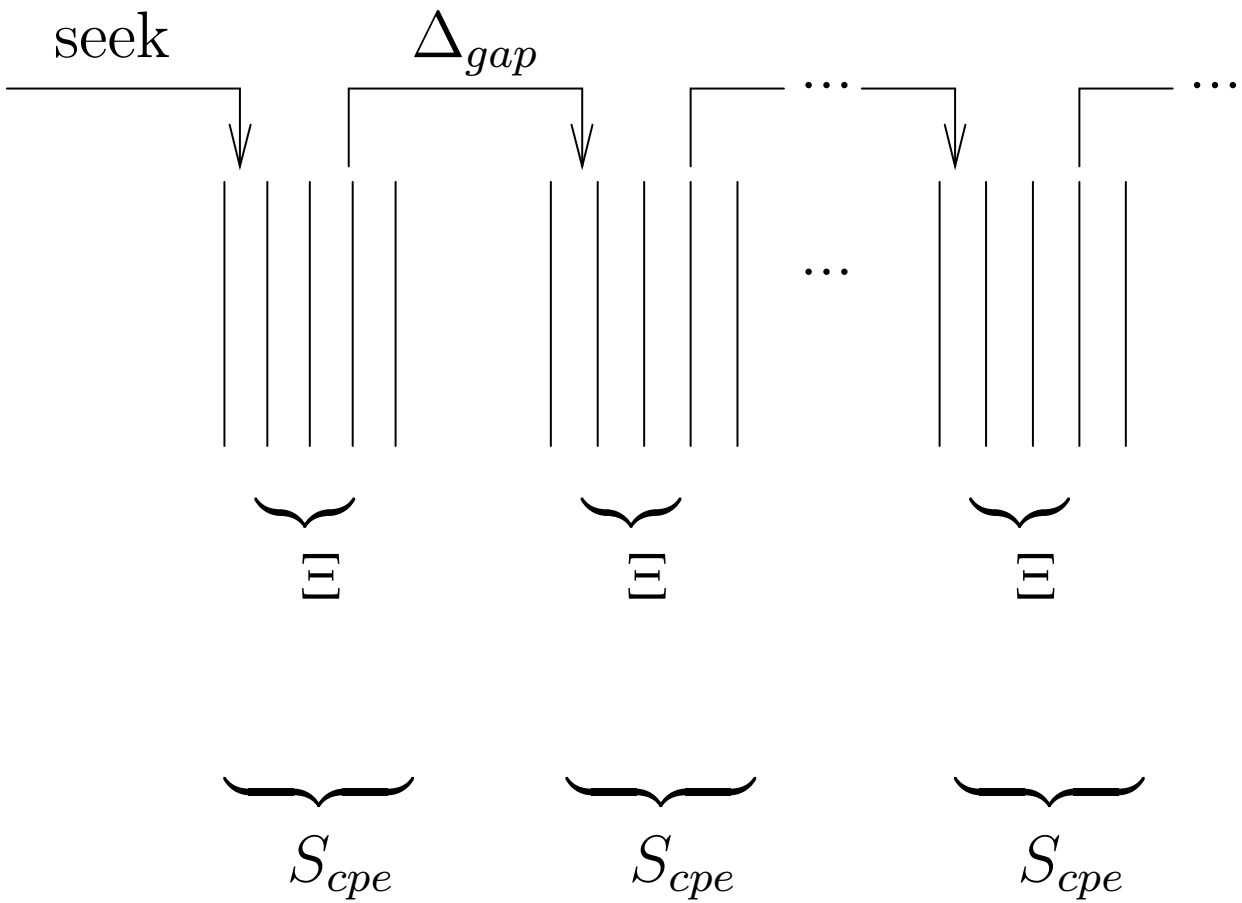
Seek Costs - Upper Bound

The first cylinder we have to visit requires a random seek with cost D_{seekavg}. (Truely upper bound: $D_{\text{fseek}}(D_{\text{cyl}} - 1)$)
After that, we have to visit the remaining $Q_c(i) - 1$ qualifying cylinders.
The segment spans a total of $S_{\text{clast}}(S_{\text{ext}}) - S_{\text{cfirst}}(1) + 1$ cylinders.
Let us assume that the first qualifying cylinder is the first cylinder and the last qualifying cylinder is the last cylinder of the segment.
Then, applying Qyang's Theorem 1 gives us the upper bound

$$C_{seek}(i) \leq (Q_c(i) - 1) D_{\text{fseek}}\left(\frac{S_{\text{clast}}(S_{\text{ext}}) - S_{\text{cfirst}}(1) + 1}{Q_c(i) - 1}\right)$$

after we have found the first qualifying cylinder.

Seek Costs - Illustration



Seek Costs - Steps

Steps:

1. Estimate for $C_{seekgap}$
2. Estimates for $C_{seekext}(i)$

Seek Costs - Interextent Costs

The average seek cost for reaching the first qualifying cylinder is D_{seekavg}. How far within the first extent are we now? We use Corollary 4 to derive that the number of non-qualifying cylinders preceding the first qualifying one in some extent i is

$$\overline{\mathcal{B}}^{S_{\text{cpe}}(i)}_{Q_c(i)} = \frac{S_{\text{cpe}}(i) - Q_c(i)}{Q_c(i) + 1}.$$

The same is found for the number of non-qualifying cylinders following the last qualifying cylinder. Hence, for every gap between the last and the first qualifying cylinder of two extents i and $i+1$, the disk arm has to travel the distance

$$\Delta_{\text{gap}}(i) := \overline{\mathcal{B}}^{S_{\text{cpe}}(i)}_{Q_c(i)} + S_{\text{cfirst}}(i+1) - S_{\text{clast}}(i) - 1 + \overline{\mathcal{B}}^{S_{\text{cpe}}(i+1)}_{Q_c(i+1)}$$

Using this, we get

$$C_{seekgap} = D_{\text{seekavg}} + \sum_{i=1}^{S_{\text{ext}}-1} D_{\text{fseek}}(\Delta_{\text{gap}}(i))$$

Seek Costs - Intraextent Costs (2)

Let us turn to $C_{seekext}(i)$. We first need the number of cylinders between the first and the last qualifying cylinder, both included, in extent i. It can be calculated using Corollary 6:

$$\Xi_{\text{ext}}(i) = \overline{\mathcal{B}}_{\text{1-span}}(S_{\text{cpe}}(i), Q_c(i))$$

Hence, $\Xi(i)$ is the minimal span of an extent that contains all qualifying cylinders.

Seek Costs - Intraextent Costs

Using $\Xi(i)$ and Qyang's Theorem 1, we can derive an upper bound for $C_{seekext}(i)$:

$$C_{seekext}(i) \leq (Q_c(i) - 1) D_{\text{fseek}}\left(\frac{\Xi(i)}{Q_c(i) - 1}\right) \tag{47}$$

Alternatively, we could formulate this as

$$C_{seekext}(i) \leq (Q_c(i) - 1) D_{\text{fseek}}\left(\overline{\mathcal{B}}^{S_{\text{cpe}}(i)}_{Q_c(i)}\right) \tag{48}$$

by applying Corollary 4.

Seek Costs - Intraextent Costs (2)

A seemingly more precise estimate for the expected seek cost within the qualifying cylinders of an extent is derived by using Theorem 8:

$$C_{seekext}(i) = Q_c(i) \sum_{j=0}^{S_{\text{cpe}}(i)-Q_c(i)} D_{\text{fseek}}(j+1)\mathcal{B}_{Q_c(i)}^{S_{\text{cpe}}(i)}(j) \qquad (49)$$

Settle Costs

The average settle cost is easy to calculate. For every qualifying cylinder, one head settlement takes place:

$$C_{settle}(i) = Q_c(i) D_{\text{rdsettle}} \tag{50}$$

Rotational Delay

Let us turn to the rotational delay.
For some given track in zone i,
we want to read the $Q_t(i)$ qualifying sectors contained in it.
On average, we would expect that the read head is ready to start reading in the middle of some sector of a track.
If so, we have to wait for $\frac{1}{2} D_{\text{zscan}}(S_{\text{zone}}(i))$ before the first whole sector ocurs under the read head.
However, due to track and cylinder skew, this event does not occur after a head switch or a cylinder switch.
Instead of being overly precise here, we igore this half sector pass by time and assume we are always at the beginning of a sector.
This is also justified by the fact that we model the head switch time explicitly.

Rotational Delay (2)

Assume that the head is ready to read at the beginning of some sector of some track.
Then, in front of us is a — cyclic, which does not matter — bitvector of qualifying and non-qualifying sectors.
We can use Corollary 5 to estimate the total number of qualifying and non-qualifying sectors that have to pass under the head until all qualifying sectors have been seen.
The total rotational delay for the tracks of zone i is

$$C_{rot}(i) = Q_t(i) \;\; D_{\text{zscan}}(S_{\text{zone}}(i)) \;\; \overline{\mathcal{B}}_{\text{tot}}(D_{\text{zspt}}(S_{\text{zone}}(i)), Q_{\text{spt}}(i))$$

where $Q_{\text{spt}}(i) = \overline{\mathcal{W}}_1^{S_{\text{sec}}, D_{\text{zspt}}(S_{\text{zone}}(i))}(N) = D_{\text{zspt}}(S_{\text{zone}}(i)) \frac{N}{S_{\text{sec}}}$ is the expected number of qualifying sectors per track in extent i. In case $Q_{\text{spt}}(i) < 1$, we set $Q_{\text{spt}}(i) := 1$.

Rotational Delay (3)

A more precise model is derived as follows.
We sum up for all j the product of (1) the probability that j sectors in a track qualify and (2) the average number of sectors that have to be read if j sectors qualify.
This gives us the number of sectors that have to pass the head in order to read all qualifying sectors.
We only need to multiply this number by the time to scan a single sector and the number of qualifying tracks.
We can estimate (1) using Theorem 7. For (2) we again use Corollary 5.

$$C_{rot}(i) = S_{\mathrm{cpe}}(i)\ D_{\mathrm{tpc}}\ D_{\mathrm{zscan}}(S_{\mathrm{zone}}(i)) * \sum_{j=1}^{\min(N, D_{\mathrm{zspt}}(S_{\mathrm{zone}}(i)))} \mathcal{X}^{S_{\mathrm{sec}}}_{D_{\mathrm{zspt}}(S_{\mathrm{zone}}(i))}(N,j)\ \overline{\mathcal{B}}_{\mathrm{tot}}(D_{\mathrm{zspt}}(S_{\mathrm{zone}}(i)), j)$$

Rotational Delay (4)

Yet another approach:
Split the total rotational delay into two components:

1. $C_{rotpass}(i)$ measures the time needed to skip unqualifying sectors
2. $C_{rotread}(i)$ that for scanning the qualifying sectors

Then

$$C_{rot} = \sum_{i=1}^{S_{\text{ext}}} C_{rotpass}(i) + C_{rotread}(i)$$

where the total transfer cost of the qualifying sectors can be estimated as

$$C_{rotread}(i) = Q_s(i) \quad D_{\text{zscan}}(S_{\text{zone}}(i))$$

Rotational Delay (5)

Let us treat the first component ($C_{rotpass}(i)$).
Assume that j sectors of a track in extent i qualify.
The expected position on a track where the head is ready to read is the middle between two qualifying sectors.
Since the expected number of sectors between two qualifying sectors is $D_{\text{zspt}}(S_{\text{zone}}(i))/j$, the expected number of sectors scanned before the first qualifying sector comes under the head is

$$\frac{D_{\text{zspt}}(S_{\text{zone}}(i))}{2j}$$

Rotational Delay (6)

The expected positions of j qualifying sectors on the same track is such that the number non-qualifying sectors between two successively qualifying sectors is the same.
Hence, after having read a qualifying sector $\frac{D_{\text{zspt}}(S_{\text{zone}}(i))}{j}$ unqualifying sectors must be passed until the next qualifying sector shows up.
The total number of unqualifying sectors to be passed if j sectors qualify in a track of zone i is

$$N_s(j,i) = \frac{D_{\text{zspt}}(S_{\text{zone}}(i))}{2j} + (j-1)\frac{D_{\text{zspt}}(S_{\text{zone}}(i)) - j}{j}$$

Rotational Delay (7)

Using again Theorem 7, the expected rotational delay for the unqualifying sectors then is

$$
\begin{aligned}
C_{rotpass}(i) &= S_{\text{cpe}}(i)\ \ D_{\text{tpc}}\ \ D_{\text{zscan}}(S_{\text{zone}}(i)) \\
&\quad * \sum_{j=1}^{\min(N, D_{\text{zspt}}(S_{\text{zone}}(i)))} \mathcal{X}^{S_{\text{sec}}}_{D_{\text{zspt}}(S_{\text{zone}}(i))}(N, j) N_s(j, i)
\end{aligned}
$$

Head Switch Costs

The average head switch cost is equal to the average number of head switches that occur times the average head switch cost.
The average number of head switch is equal to the number of tracks that qualify minus the number of cylinders that qualify since a head switch does not occur for the first track of each cylinder.
Summarizing

$$C_{headswitch} = \sum_{i=1}^{S_{\text{ext}}} (Q_t(i) - Q_c(i)) \quad D_{\text{hdswitch}} \tag{51}$$

where Q_t is the average number of tracks qualifying in an extent.

Discussion

We neglected many problems in our disk access model:

- partially filled cylinders,
- pages larger than a block,
- disk drive's cache,
- remapping of bad blocks,
- non-uniformly distributed accesses,
- clusteredness,
- and so on.

Whereas the first two items are easy to fix, the rest is not so easy.

Selectivity Estimations

- previous slides assume that we "know" how many tuples qualify
- but this has to be estimated somehow
- similar for join ordering algorithms etc.
- cardinalities (and thus selectivities) are fundamental for query optimization
- we will now look at deriving some estimations

Examples

SQL examples for typical selectivity problems:

- **select** *
  from rel r
  where r.a=10
- **select** *
  from rel r
  where r.b>2
- **select** *
  from rel1 r1,rel2 r2
  where r1.a=r2.b

The different problems require different approaches.

Heuristic Estimations

Some commonly used selectivity estimations:

predicate	selectivity	requirement
$A = c$	$1/\|D(A)\|$	if index on A
	$1/10$	otherwise
$A > c$	$(\max(A) - c)/(\max(A) - \min(A))$	if index on A, interpol.
	$1/3$	otherwise
$A_1 = A_2$	$1/\max(\|D(A_1)\|, \|(D(A_2)\|)$	if index on A_1 and A_2
	$1/\|D(A_1)\|$	if index on A_1 only
	$1/\|D(A_2)\|$	if index on A_2 only
	$1/10$	otherwise

Note: Without further statistics, $|D(A)|$ is typically only known (easily estimated) if A is a key or there is an index on A.

Using Histograms

- selectivity can be calculated easily by looking at the real data
- not feasible, therefore look at aggregated data
- histograms partition the data values into buckets

A histogram $H_A : B \rightarrow \mathbb{N}$ over a relation R partitions the domain of the aggregated attribute A into disjoint buckets B, such that

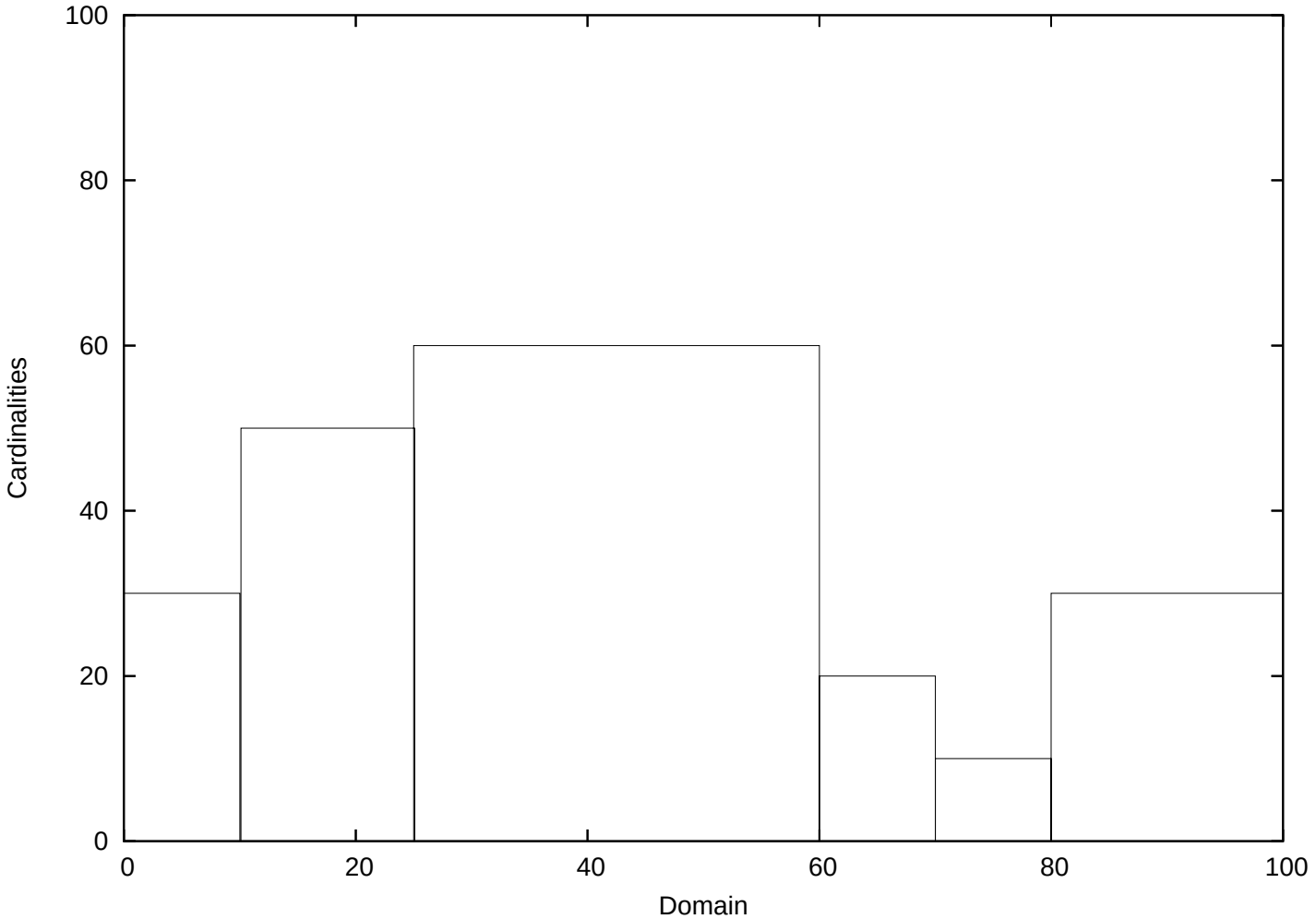
$$H_A(b) = |\{r | r \in R \wedge R.A \in b\}|$$

and thus $\sum_{b \in B} H_A(b) = |R|$.

Choosing B is very important, as we will see on the next slides.

Using Histograms (2)

A rough histogram might look like this:



Using Histograms (3)

Given a histogram, we can approximate the selectivities as follows:

$$A = c \qquad \frac{\sum_{b \in B : c \in b} H_A(b)}{\sum_{b \in B} H_A(b)}$$

$$A > c \qquad \frac{\sum_{b \in B : c \in b} \frac{\max(b) - c}{\max(b) - \min(b)} H_A(b) + \sum_{b \in B : \min(b) > c} H_A(b)}{\sum_{b \in B} H_A(b)}$$

$$A_1 = A_2 \qquad \frac{\sum_{b_1 \in B_1, b_2 \in B_2, b' = b_1 \cap b_2 : b' \neq \emptyset} \frac{\max(b') - \min(b')}{\max(b_1) - \min(b_1)} H_{A_1}(b_1) \frac{\max(b') - \min(b')}{\max(b_2) - \min(b_2)} H_{A_2}(b_2)}{\sum_{b_1 \in B_1} H_{A_1}(b_1) \sum_{b_2 \in B_2} H_{A_2}(b_2)}$$

Using Histograms - Remarks

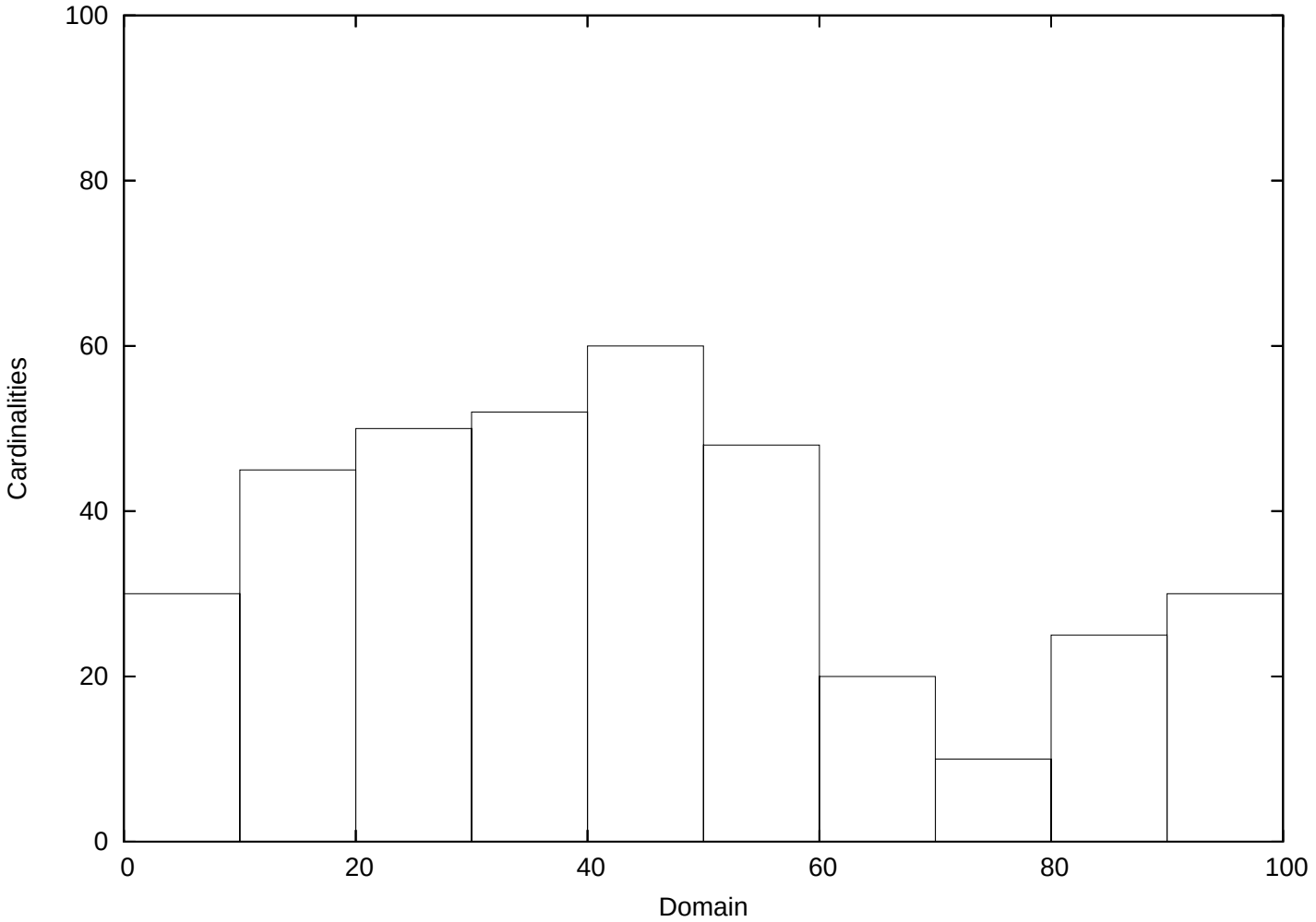
- estimations on previous slide can be improved
- in particular, the $A = c$ case is only a rough approximation
- requires more information
- if we interpret the histogram as a density function, $P(A = c) = 0$!
- a reasonable upper bound, though
- the $A > c$ case is more sound
- $A_1 = A_2$ assumes independence etc.

Building Histograms

- the buckets chosen greatly affect the overall quality
- histogram does not discern items within one bucket
- therefore: try to put items into different buckets
- how to choose the buckets?
- typical constraint: histogram size. n buckets (fixed)
- for a given set of data items, find a good histogram with n buckets
- additional constraint: data distribution is unknown (real data)

Building Histograms - Equiwidth

Partitions the domain into buckets with a fixed width



Building Histograms - Equiwidth (2)

Advantages:

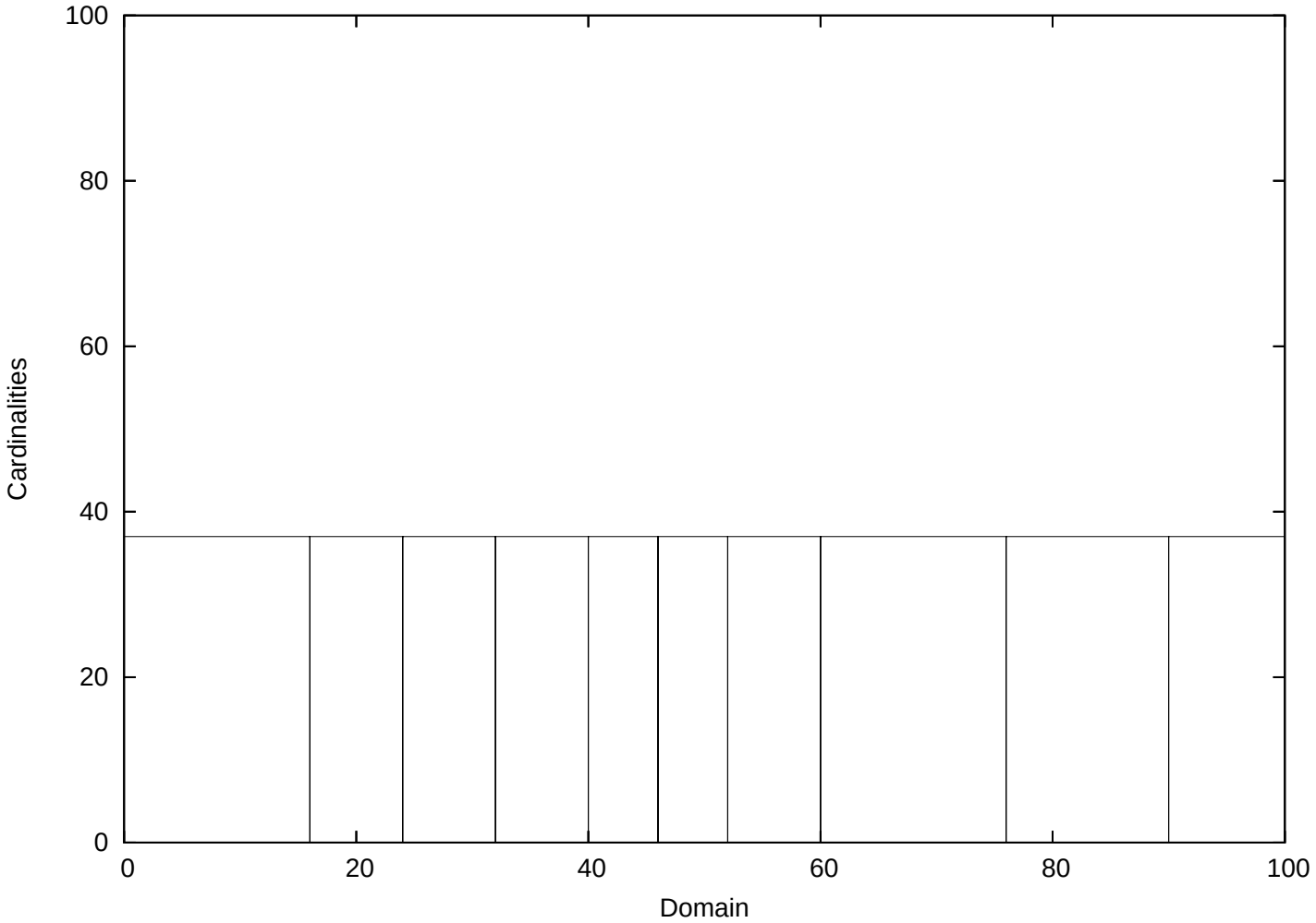
- easy to compute
- bucket boundaries can be computed (require no space)

Disadvantages:

- samples the domain uniformly
- does not handle skewed data well
- skew can lead to very uneven buckets
- greater estimation error in large buckets
- particular bad for zipf-like distributions

Building Histograms - Equidepth

Chooses the buckets to contain the same number of items



Building Histograms - Equidepth (2)

Advantages:

- adopts to data distribution
- reduces maximum error

Disadvantages:

- more involved (sort or similar)
- both boundaries and depth have to be stored (ties)

Very common histogram building technique

Building Histograms - Interpolation

- data is usually not completely random
- can we increase accuracy by interpolation?
- either within buckets (common) or instead of buckets (uncommon)
- histogram is a density function, not continuous, hard to interpolate
- use the equivalent distribution function instead
- very good for estimating $A > c$

Discussion

- estimations more complex in practice
- potentially different goals: maximum vs. average error
- histograms for derived values
- histogram convolution
- handling correlations
- multi-dimensional histograms
- cardinality estimators (sketches, MIPS etc.)