ELSEVIER

Data & Knowledge Engineering 14 (1994) 251–264

DATA & KNOWLEDGE ENGINEERING

Clustering factor estimation for totally clustered attributes

Fabio Grandi, Maria Rita Scalas

C.I.O.C.-C.N.R. and Dipartimento di Elettronica Informatica e Sistemistica, Università di Bologna, Bologna, Italy

Received 4 February 1994; accepted in revised form 25 July 1994; accepted 27 September 1994

Abstract

Cost models based on the *clustering factor* (*CF*) of the attributes have been proposed and shown to be attractive for block access estimation in databases, thanks to their accuracy and economy of use. While query optimizers can use the actual *CF*s, measured from the data, physical design methods and tools must rely on estimates before the data are stored.

In this paper we present a *CF* estimation procedure which can be applied to *totally clustered* attributes (e.g. ordered attributes). Simple and accurate approximations of the derived formulas are also introduced.

Simulations show the accuracy of the proposed *CF* estimates and the improvement in their behaviour compared to previously published estimates. Reliability for physical design of cost models based on the *CF* in the presence of a skewed data distribution is also discussed.

Keywords: Databases; Clustering factor; Performance evaluation; Physical design; Relational database

1. Introduction

Cost models for data access by an index are required in relational databases for query optimization [13] and physical design [9]. Usage and maintenance of models based on the *clustering factor* [2] are not expensive, because they are based on a single parameter value per attribute and can easily be embedded in the systems and design tools already in use. Cost models for data access based on this parameter give results that are encouraging for their accuracy, economy of use and applicability in a wide range of situations [1,2,5–7,10–12] with respect to more sophisticated models.

The clustering factor (*CF*) of an attribute takes into account the actual placement of the different values of an attribute on the data pages, by counting the average number of tuples with the same value on a page. The average is evaluated over the values of the attribute and over the pages. If we define as *cluster* the set of occurrences of the same attribute value on a page, the *CF* represents the average size of a cluster. Notice that the term *clustering* is used to mean the presence of more than one tuple with the same value of an attribute on the same

SSDI: 0169-023X(94)00026-3

TID	A_1	A_2	A_3	A_4
1.1	a	X	μ	Ξ
1.2	a	X	μ	Α
1.3	a	T	μ	Α
1.4	a	T	θ	Γ
1.5	a	T	θ	Γ
1.6	b	Y	θ	Γ
2.1	b	Y	θ	N
2.2	b	S	θ	Π
2.3	b	S	φ	Π
2.4	b	U	φ	M
2.5	b	U	φ	Σ
2.6	d	U	φ	Σ
3.1	d	U	φ	Σ
3.2	d	U	φ	E
3.3	d	U	φ	R
3.4	f	V	ω	T
3.5	f	V	ω	Ψ
3.6	f	V	λ	X

TID	A_1	A_2	A_3	A_4
4.1	f	V	λ	H
4.2	f	W	λ	Λ
4.3	g	W	λ	Λ
4.4	g	W	λ	Z
4.5	g	R	λ	Φ
4.6	g	R	λ	Φ
5.1	g	R	λ	K
5.2	g	Q	ψ	Θ
5.3	i	Q	ψ	Θ
5.4	i	O	ψ	B
5.5	i	O	ψ	Υ
5.6	i	O	ϵ	Ω
6.1	i	Z	ϵ	Ω
6.2	i	Z	ϵ	Ω
6.3	k	Z	ϵ	I
6.4	k	N	ϵ	Δ
6.5	k	N	ξ	P
6.6	k	P	ξ	O

Fig. 1. This figure shows the six pages of a sample relation. All attributes A_1, A_2, A_3 and A_4 are *totally clustered* (attribute A_1 is also *ordered*). Average duplications and clustering factors are: 5.14 and 3 for A_1; 2.77 and 2.12 for A_2; 4.5 and 2.77 for A_3; 1.5 and 1.33 for A_4. The column TID (not belonging to the relation) contains the Tuple IDentifiers in the format 'PID. *Offset-in-the-page*'.

page. *Total clustering* means all the tuples with the same value of the attribute are consecutive in the storage space and *ordering* is a special case of total clustering where the values are also sorted. Fig. 1 shows examples of totally clustered attributes.

Two cases can be distinguished in the use of the clustering factor: the operative phase, when the data are already stored in the database and the actual *CF* can easily be measured on the data (or on the index), and the design phase [6], when data are not yet available, and the *CF* can be estimated by means of the known or forecast parameter values (i.e. number of pages, tuples and distinct key values). The knowledge of the CF can be used in this phase to evaluate the convenience of building an index on an attribute. The term *key* is used in the following to indicate an attribute on which an index is built: such keys are the distinct attributes which will be used as search keys in data retrieval.

In this work we present a simple method of estimating, during the physical design process, the clustering factor of totally clustered attributes in relations with a uniform number of tuples per page and without assumptions on the distribution of key values in the tuples. The rest of the paper is organized as follows. A formal definition of the clustering factor and an approximate estimation of its value for totally clustered attributes can be found in Section 2. Section 3 concerns the derivation of our new estimate, which is experimentally validated in Section 4. A discussion on the applicability of our estimate in the presence of skewed attributes is also included. Conclusions are finally reported in Section 5. A list of the symbols used in the paper can be found in Table 1.

2. Previous work

The clustering factor was originally defined in [2] as:

Table 1
A synopsis of the symbols

NT	number of tuples
NP	number of pages
NK	number of distinct key values
TP	average number of tuples per page
KP	average number of distinct key values per page
DK	average duplication of key values
CF	average clustering factor of the key attribute
PID	page identifier in an index leaf
$NPID$	total number of PIDs in the index
$PID(K_i)$	number of PIDs in the index associated with key K_i
$PID(P_i)$	number of PIDs in the index referring to page P_i
	with the constraints: $NT = NP \cdot TP = NK \cdot DK$, $KP \cdot CF = TP$

$$CF = \frac{NT}{NPID} \tag{1}$$

where *NPID* is the total number of Page IDentifiers (PIDs) – *not necessarily distinct* – which can be found in the leaves of an index built on that attribute, if the index leaves contain $\langle key\colon PID_1, \ldots, PID_m \rangle$ groups, with only distinct PIDs stored within each group [2]. More in general, indexes can be assumed to be standard B^+-trees or similar [8], with leaves containing $\langle key\colon TID_1, \ldots, TID_n \rangle$ groups, where a TID is a Tuple IDentifier composed of a PID and an offset. In this case, in order to compute *NPID*, the same PIDs are considered equal within the same group but *distinct* when occurring in different groups.

2.1. An approximate estimation

A first estimate for physical design of the clustering factor for a totally clustered attribute was proposed by Ciaccia and Maio in [6]. They used a 'normalized' clustering factor *cf* defined as:

$$cf = \frac{NP}{NPID} \tag{2}$$

Its reciprocal, cf^{-1}, represents the average number of clusters per page, that is the average number of distinct key values per page (*KP*). From definitions (1) and (2) we can easily obtain:

$$CF = TP \cdot cf \tag{3}$$

In [6], the estimate of *cf* for an ordered attribute is computed as:

$$cf = \begin{cases} NP/NK & \text{if } NK > NP \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

Obviously, the validity of (4) is not really limited to the case of ordering but extends to the general case of total clustering. Total clustering without ordering occurs, for instance, for an attribute with a 1:1 functional dependency on the primary attribute (on which the relation is sorted). By means of $NT = TP \cdot NP = DK \cdot NK$, Eq. (4) can be rewritten as:

$$cf = \begin{cases} DK/TP & \text{if } DK < TP \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

Using (3), we can finally derive the Ciaccia–Maio estimate, say CF_0, of the clustering factor for a totally clustered attribute:

$$CF_0 = \begin{cases} DK & \text{if } DK \leqslant TP \\ TP & \text{if } TP \leqslant DK \end{cases} = \min \{TP, DK\} \tag{6}$$

The derivation of (6) can easily be explained. If $DK \leqslant TP$, more than one key is contained in a page and a mean estimate of the number of distinct key values per page is given by $KP = TP/DK$. If $TP \leqslant DK$, a key value spans more than one page, thus in a given page only one key value is contained: $KP = 1$. Eq. (4) immediately follows ($cf = 1/KP$), whereas (6) follows from $CF = TP/KP$.

The shortcoming in (6) is that the estimates used for KP are not globally correct, due to a sort of border effect. Let us consider, for instance, the case $TP \leqslant DK$. When any key value spans more than one page, some pages of the relation contain exactly one key value, but all the other pages contain two key values, unless the key value change always occurs at a page boundary. This can only happen if all the key values have a number of duplicates which are exact multiples of TP. An analogous effect can be observed when $DK \leqslant TP$. In this case some key values are completely contained in one page, whereas the others exactly span two pages. For instance, attribute A_1 in Fig. 1 (with $DK = 5.14 < TP = 6$) has only values $\{a, k\}$ completely contained in a page and values $\{b, d, f, g, i\}$ span two pages.

3. A more precise estimation

In this section we derive a more accurate estimate of CF in the presence of total clustering, taking into account the border *effect* described above. The value $NPID$ to be used in the CF definition (1), can be computed as

$$NPID = \sum_{i=1}^{NK} PID(K_i) \tag{7}$$

where $PID(K_i)$ represents the number of distinct $PIDs$ associated to the key value K_i ($PID(K_i) \geqslant 1$ when $DK \geqslant TP$), or as

$$NPID = \sum_{i=1}^{NP} PID(P_i) \tag{8}$$

where $PID(P_i)$ represents the total number of PIDs referencing the same page P_i ($PID(P_i) \geqslant 1$ when $TP \geqslant DK$). The two evaluation methods that follow from (7) and (8) are the subject of the following subsections. Their results will be generalized in Section 3.3.

3.1. *Estimation for high duplication* ($DK \geq TP$)

Since $DK \geq TP$, we can evaluate *NPID* by means of (7). In this case all the duplicates of the key value K_i span one or more consecutive pages and $PID(K_i)$ is exactly the number of such pages.

Let us number from 1 to *NK* the key values as they are consecutively found in the relation, but not necessarily in their value order. Thus, K_{i-1} physically precedes K_i ($i = 2, \ldots, NK$) in the relation, but $K_i < K_{i+1}$ only if the relation is ordered. Let p_i^F and p_i^L be the page numbers of the first and the last page spanned by the key K_i, respectively. Obviously, we have $p_1^F = 1$ and $p_{NK}^L = NP$. With these positions, we have:

$$PID(K_1) = p_1^L \tag{9}$$

$$PID(K_i) = p_i^L - p_i^F + 1 \quad (i = 2, \ldots, NK) \tag{10}$$

Moreover, if q_i^K represents the probability that the first occurrence of the key value K_i is placed on the *first position* of the first page spanned by such key, we have (for $i = 2, \ldots, NK$):

$$p_i^F = \begin{cases} p_{i-1}^L + 1 & \text{with probability } q_i^K \\ p_{i-1}^L & \text{with probability } 1 - q_i^K \end{cases} \tag{11}$$

where the first case accounts for the possibility that the key change between K_{i-1} and K_i occurs at a page boundary. Therefore, the mean value of p_i^F is given by

$$E[p_i^F] = (p_{i-1}^L + 1)q_i^K + p_{i-1}^L(1 - q_i^K) = p_{i-1}^L + q^K{}_i \tag{12}$$

By substituting (12) for p_i^F into (10), we have the following estimate:

$$PID(K_i) = p_i^L - p_{i-1}^L + 1 - q_i^K \quad (i = 2, \ldots, NK)$$

and, thus, using also (9), (7) becomes:

$$\begin{aligned} NPID &= p_1^L + \sum_{i=2}^{NK}(p_i^L - p_{i-1}^L + 1 - q_i^K) \\ &= p_{NK}^L + \sum_{i=2}^{NK}(1 - q_i^K) \\ &= NP + (NK - 1)(1 - q^K) \end{aligned}$$

where q^K is the average value of the probabilities $q_2^K, \ldots, q_{NK}^K$. Such a value, representing the average probability that the first occurrence of a given key value be placed on the first position of the first page spanned by such key value, can be simply estimated as

$$q^K = 1/TP \tag{14}$$

as TP represents the average number of positions in a page. By virtue of value (14), Eq. (13) yields

$$NPID = NP + (NK - 1)(1 - 1/TP) \tag{15}$$

and, finally,

$$CF = \frac{NT}{NP + (NK - 1)(1 - 1/TP)} \tag{16}$$

On multiplying and dividing by $TP \cdot DK/NT$, we can also write:

$$CF = \frac{TP \cdot DK}{TP + DK + \frac{TP \cdot DK}{NT} - \frac{NK - 1}{NK}} \tag{17}$$

If NK is not too small, we can derive our first estimate, say CF_1, of the clustering factor for a totally clustered attribute:

$$CF \simeq CF_1 = \frac{TP \cdot DK}{TP + DK + TP \cdot DK/NT - 1} \tag{18}$$

3.2. Estimation for low duplication ($TP \geq DK$)

Using Eq. (8), when $TP \geq DK$, yields a perfectly dual derivation procedure in which tuples spanned by a page or by a key value exchange the role played. As a matter of fact, letting k_i^F and k_i^L be the key numbers contained in the first and in the last position of the page P_i ($k_1^F = 1$ and $k_{NP}^L = NK$), respectively, we have:

$$PID(P_1) = k_1^L \tag{19}$$

$$PID(P_i) = k_i^L - k_i^F + 1 \quad (i = 2, \ldots, NP) \tag{20}$$

$$k_i^F = \begin{cases} k_{i-1}^L + 1 & \text{with probability } q_i^P \\ k_{i-1} & \text{with probability } 1 - q_i^P \end{cases} \tag{21}$$

$$E[k_i^F] = k_{i-1}^L + q_i^P \tag{22}$$

$$\begin{aligned} NPID &= k_1^L + \sum_{i=2}^{NP} (k_i^L - k_{i-1}^L + 1 - q_i^P) \\ &= NK + (NP - 1)(1 - q^P) \end{aligned} \tag{23}$$

where q_i^P represents the probability that the first position of the page P_i contains the *first duplicate* of the first key value this page contains, with average value on the pages q^P evaluated as $1/DK$, since DK represents the average number of duplicates of a key value. Therefore, we have:

$$NPID = NK + (NP - 1)(1 - 1/DK) \tag{24}$$

$$CF = \frac{TP \cdot DK}{TP + DK + \frac{TP \cdot DK}{NT} - \frac{NP - 1}{NP}} \tag{25}$$

$$\approx \frac{TP \cdot DK}{TP + DK + TP \cdot DK/NT - 1} \tag{26}$$

where the last approximation is immaterial if NP is not too small.

The last derived CF is the same as CF_1 defined by (18), as can also be argued from the structure of the formula (18), which is symmetrical with respect to TP and DK.

3.3. *A generalized estimation formula*

Since we obtained – with good approximation – the same value either when $DK \geqslant TP$ or when $DK \leqslant TP$, a further simplification consists in assuming that value to be valid in the most general case. For instance, very skewed distributions of key values may give rise to $DK_i < TP$ for some key values and to $DK_i > TP$ for other key values. Zipf distribution [16] is a clear example of this case. Our hypothesis is to assume the CF_1 value (18) as a good estimate of the average clustering factor in any case. Experimental results for the evaluation of this assumption are provided in the next section.

By applying identity $KP = TP/CF$ to (18), for a totally clustered attribute we also have:

$$KP = \frac{DK + TP - 1}{DK} + \frac{1}{NP} \tag{27}$$

If $TP + DK$ and NP are large, we can approximate (27) as:

$$KP \approx 1 + \frac{TP}{DK} \tag{28}$$

which shows the contribution of the *border effect* on the clustering. Since $KP = cf^{-1}$, when $DK \leqslant TP$ we have from (5) $KP \approx TP/DK$, thus '1' is the border contribution in (28); when $TP \leqslant DK$ Eq. (5) yields $KP \approx 1$ and thus the border contribution is 'TP/DK.'

4. Experimental validation

In this section we analyze the accuracy of the CF estimate (18):

$$CF_1 = \frac{DK \cdot TP}{DK + TP + DK \cdot TP/NT - 1}$$

Moreover, we consider the following two approximations of CF_1 and evaluate their accuracy:

$$CF_2 = \frac{DK \cdot TP}{DK + TP - 1} \tag{29}$$

$$CF_3 = \frac{DK \cdot TP}{DK + TP} \tag{30}$$

The last approximation can easily be remembered as it is the harmonic mean between *DK* and *TP* and it can be directly derived by neglecting the possibility that a key value change and a page change may coincide ($q^K \simeq q^P \simeq 0$). The accuracy of CF_1, CF_2 and CF_3 estimates is also matched against prediction errors of the Ciaccia–Maio CF_0 estimate (6).

In the relations used for simulations, the duplication of key values and the number of tuples in the pages are generated as random variables. In particular, two classes of experiments have been effected: in the former key values have been drawn from a uniform distribution, in the latter from a Zipf distribution. The *DK* and *TP* values used in the estimation formulae are the average values computed from the actual distributions. The non-constant value of *TP* corresponds to a different number of tuples per page (as resulting from variable-length records).

The figures from Fig. 2 to Fig. 5 show the *CF* values for a totally clustered attribute in relations with 30,000 tuples. Three graphs are displayed in every figure, even if two of them in practice coincide: the piecewise-linear graph plots the CF_0 estimate provided by (6), whereas the other two curves plot the actual values of *CF* measured from the two classes of experiments (uniform and Zipf distribution of attribute values). Eight types of relations have been studied: for each of the two classes of key distributions, four numbers of pages have been considered. The number of pages is 2,500 in Fig. 2 ($TP = 12 \pm 3.46$), 375 in Fig. 3 ($TP = 80 \pm 8.93$), 200 in Fig. 4 ($TP = 150 \pm 12.22$) and 75 in Fig. 5 ($TP = 400 \pm 19.87$). In all the simulations, the number of key values *NK* is the independent variable, which ranges from 100 (75 for the fourth relation) to 30,000. However, $DK = NT/NK$ is reported for greater clarity as abscissa in the figures. The approximation error of (6) is apparent in the figures, whereas all the estimates CF_1, CF_2 and CF_3 yield values which would be indistinguishable from the actual *CF*s at the scale of the figures.

Table 2 resumes the *maximal* percentage errors ($E_i = 100(CF_i - CF)/CF$, $i = 0, 1, 2, 3$) provided by the estimates, measured on a run of three simulations per relation type. The

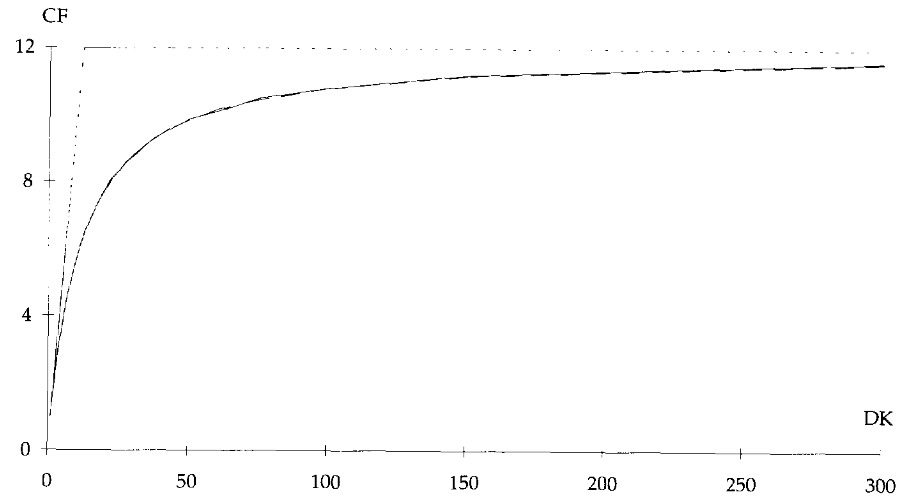


Fig. 2. Experimental CF for a totally clustered attribute following uniform and Zipf distribution in a 30 000-tuple 2500-page relation.

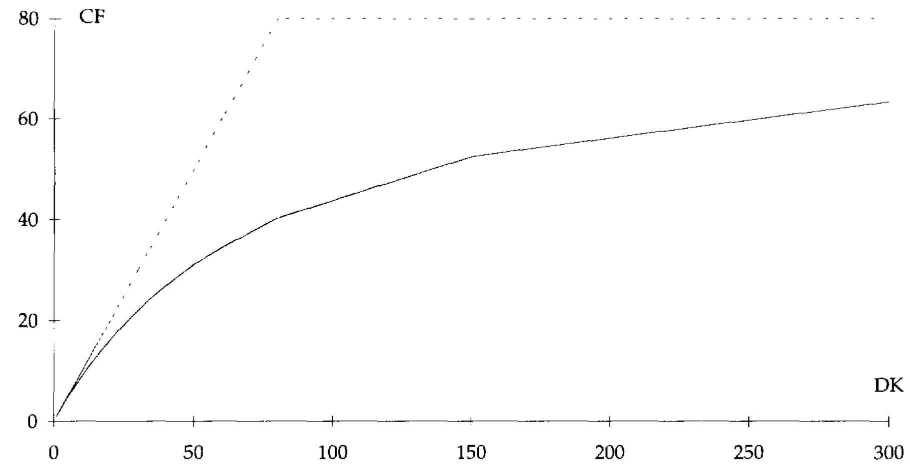


Fig. 3. Experimental CF for a totally clustered attrinute following uniform and Zipf distribution in a 30 000-tuple 375-page relation.

worst case for (6) occurs for $DK = TP$. The *border effect*, which is null for $DK = 1$ or $DK = NT$, is maximum for $DK = TP$. In this case, Eq. (6) yields $CF = TP$, whereas Eq. (18) yields

$$CF = \frac{TP^2}{2 \cdot TP - 1 + TP^2/NT} \approx \frac{TP}{2}$$

(if $TP < NP$ then $TP^2/NT < 1$) which is an accurate value. The maximum error of (6) is therefore about 100%, as confirmed by Table 2. On the other hand, estimates CF_1 and CF_2 are proved to be very good, as they provide a maximal error less than 1% in all the experiments, in the presence of both uniform and Zipf distributed data.

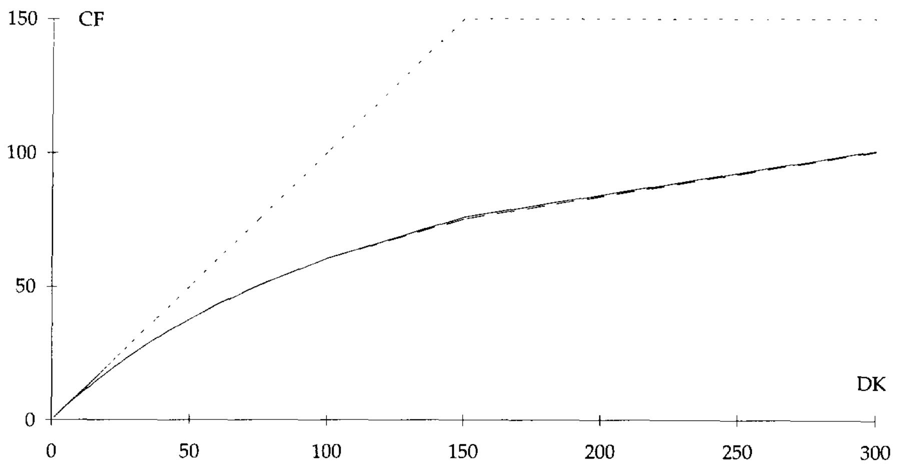


Fig. 4. Experimental CF for a totally clustered attribute following uniform and Zipf distribution in a 30 000-tuple 200-page relation.

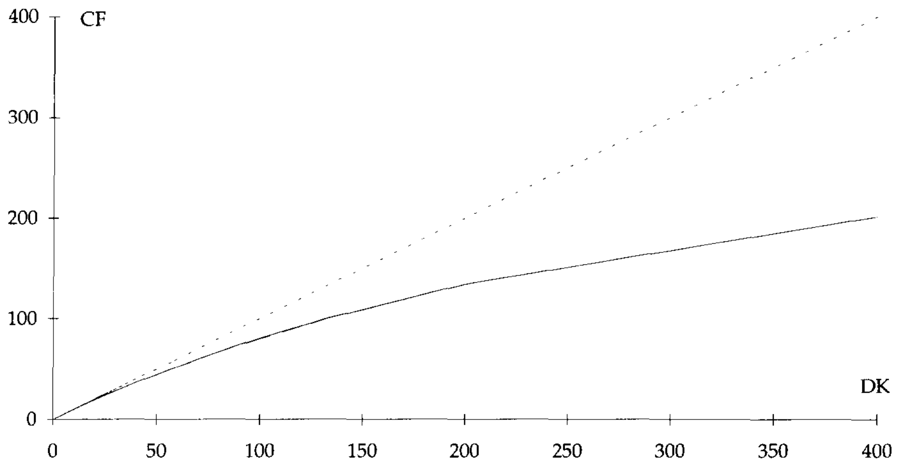


Fig. 5. Experimental CF for a totally clustered attrinute following uniform and Zipf distribution in a 30 000-tuple 75-page relation.

4.1. *On the estimate applicability*

Simulation results in the previous Section have shown that all the estimation formulas CF_1, CF_2 and CF_3 provide accurate estimations of the clustering factor for a totally clustered attribute, both in the presence of uniform or skewed distributions of key values. Therefore, database design tools using such estimates can actually rely on precise CF values even in the absence of measurements on the data. However, the application of cost models based on the knowledge of the clustering factor to totally clustered attributes needs some caution.

If attribute values follow a uniform distribution, exact cost models based on the clustering factor [5,10] can be used. For instance, the number HP of pages hit by a query referencing HK distinct key values can be estimated as:

$$HP(HK) = NP\left[1 - \frac{\binom{NK - KP}{HK}}{\binom{NK}{HK}}\right]$$

This model is based on *uniform clustering*, that is, the distinct number of key values per page (KP) is assumed to be constant over the pages. In general, this assumption does not require attribute values to be either uniformly distributed or randomly placed over the pages. On the contrary, when the attribute is totally clustered the assumption entails a uniform distribution over the domain (i.e. equifrequency of values).

Table 2
Maximum percentage error of the CF estimates

NP	Uniform				Zipf			
	E_1	E_2	E_3	E_0	E_1	E_2	E_3	E_0
2,500	0.68	0.69	7.56	91.39	0.67	0.70	7.69	92.52
375	0.48	0.52	1.54	98.67	0.52	0.47	1.47	98.40
200	0.67	0.92	1.25	97.50	0.40	0.60	1.00	98.50
75	0.34	0.60	0.80	98.67	0.34	0.60	0.80	98.67

In general, if KP_i is the number of distinct key values stored in page P_i, then the probability that page P_i is referenced by the query is given by:

$$\frac{\binom{NK - KP_i}{HK}}{\binom{NK}{HK}}$$

Therefore, the expected value of the total number of pages referenced by the query can be evaluated [6] as:

$$HP(HK) = \sum_{i=1}^{NP} \left[1 - \frac{\binom{NK - KP_i}{HK}}{\binom{NK}{HK}} \right] \tag{32}$$

For queries referencing only one key value, the models based on uniform clustering are exact. As a matter of fact, Eq. (32) reduces to:

$$HP(1) = NP\frac{KP}{NK} = \frac{DK}{CF} \tag{33}$$

where

$$KP = \sum_{i=1}^{NP} KP_i / NP \tag{34}$$

is still the average number of key values per page. However, for queries referencing more than one key value and when KP_i is not constant in every page, it can be shown [4] that the exact expression (32) is majorized by (31), with KP given by (34). In other words, when KP_i distribution is very skewed over the pages (as in the case of data following Zipf's law), the model (31) based on a uniform clustering factor provides an overestimation of the actual costs. The case study that follows is introduced in order to highlight the relevance of such an overestimation.

We used for query simulations four 30,000-tuple 200-page relations with varying degrees of clustering of a totally clustered attribute. In particular, we used the values 100, 500, 1000, and 5,000 for the number of distinct key values NK. We ran five groups of queries per relation (each group retrieving a number of key values ranging from 1 to NK) and registered actual average access costs. Actual costs were matched against expected costs provided by formula (31) (used with CF_1 estimate) and percentage prediction errors were computed. Actual costs were also matched against expected costs provided by a linear model and a uniform model. The linear model assumes a ratio of accessed pages equal to the ratio of referenced key values, that is:

$$HP(HK) = \frac{HK}{NK} NP$$

and is the cost model generally used by optimizers and design tools for the retrieval by an index of a sorted attribute. The uniform model used is the 'classical' Yao's formula [15] based on total uniformity (i.e. uniformity of data values and random placement on the pages).

Table 3
Estimation errors of cost models for sorted Zipf attributes

NK	*CF* model		Linear model		Uniform model	
	Maximum % error	Average % error	Maximum % error	Average % error	Maximum % error	Average % error
100	39.28	8.97	27.37	14.34	5,895	316.71
500	53.62	14.54	154.20	88.38	2,190	117.05
1,000	48.43	26.91	135.81	97.27	2,694	98.60
5,000	60.13	19.20	107.44	99.66	492.6	29.31

Comparison with the uniform model is included for reference because it would be chosen by current design tools in the presence of totally clustered *but not sorted* attributes, as for example attributes with a 1:1 functional dependency on the sorting attribute.

Table 3 summarizes the results of the experiments. In particular, the column 'Average % Errors' contains a further average of the errors with respect to the number of retrieved key values. These values can be used as merit figures for the global behaviour of the *CF* cost model for physical design (global predictive power rather than adherence to a particular query is privileged). It can be seen that, although the average percentage error of formula (31) can exceed 20%, it outperforms the linear and the uniform models which are usually embedded in current design tools. Cost models based on uniformity are useless for estimating access costs in the presence of totally clustered and skewed attributes, as is clearly shown in the Table.

However, access costs estimated using the *CF* model are significant even though not very precise also in the presence of very skewed data distributions. The methodology outlined in [6] could be used to achieve a good database design (even if not the optimum). On the other hand, better cost estimates could only be obtained at the expense of the management of much more complex cost models [3,14]. Moreover, such models rely on detailed knowledge of data distribution (e.g. KP_i histograms) that may not be available in the design phase.

5. Conclusions

In this paper we have presented simple estimation formulas for the clustering factor of a totally clustered attribute to be used for database physical design.

By means of simulation experiments we have shown that the new estimates are better than the previously published Ciaccia–Maio estimate. Indeed our formula CF_1 and its approximations CF_2 and CF_3 are very accurate even in the presence of highly skewed distributions of key values. In particular, the maximal prediction error provided by CF_1 and CF_2 never reached 1% in all the effected simulation trials.

The rationale of the improvement introduced with respect to the Ciaccia–Maio model consists in having taken into account the 'border effect' due to the fact that page boundaries and change of key values very seldom coincide along the storage space.

Further experiments have been done to analyze the reliability for physical design of access cost models based on the *CF* for skewed data. They have shown that even though access costs may be overestimated (with a percentage error exceeding 20%), their indications are however significant and more precise than those provided by alternative and widely used models.

Future work should be devoted to the study of the 'border effect' also in secondary clustering, that is to the estimation of the clustering factor of an attribute with a known functional dependency on another attribute whose clustering factor is known (or can be reliably estimated). As shown in [6], this would provide a very powerful physical design methodology based on the clustering factors. To this purpose, new formulas should be derived in order to improve the estimates proposed by Ciaccia and Maio, which do not consider the 'border effect'.

References

[1] S. Bergamaschi and M.R. Scalas, Choice of the optimal number of blocks for data access by an index, *Informat. Syst.* 11(3) (1986) 199–209.

[2] F. Bonfatti, D. Maio, M. Spadoni and P. Tiberio, An indexing technique for relational data bases, *Proc. IEEE COMPSAC* (Chicago, 1980) 784–791.

[3] S. Christodoulakis, Estimating block selectivities, *Informat. Syst.* 9(1) (1984) 69–79.

[4] S. Christodoulakis, Implications of certain assumptions in database performance evaluation, *ACM Trans. Database Syst.* 9(2) (June 1984) 163–196.

[5] P. Ciaccia, Block access estimation for clustered data, *IEEE Trans. Knowledge and Data Engrg.* 5(4) (Aug. 1993) 712–718.

[6] P. Ciaccia and D. Maio, Access cost estimation for physical database design, *Data & Knowledge Engrg.* 11 (1993) 125–150.

[7] P. Ciaccia and M.R. Scalas, Optimization strategies for relational disjunctive queries, *IEEE Trans. Software Engrg.* 15(10) (Oct. 1989) 1217–1235.

[8] D. Comer, The Ubiquitous B-tree, *ACM Comput. Surv.* 11(2) (June 1979) 121–137.

[9] S. Finkelstein, M. Schkolnick and P. Tiberio, Physical database design for relational databases, *ACM Trans. Database Syst.* 13(1) (March 1988) 91–128.

[10] F. Grandi and M.R. Scalas, Block access estimation for clustered data using a finite LRU buffer, *IEEE Trans. Software Engrg.* 19(7) (July 1993) 641–660.

[11] D. Maio, M.R. Scalas and P. Tiberio, Dynamic non-dense indexes in relational databases, *Informat. Syst.* 9(3/4) (1984) 207–216.

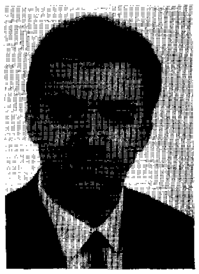
[12] D. Maio, M.R. Scalas and P. Tiberio, On estimating access costs in relational databases, *Inform. Process. Lett.* 19(3) (Oct. 1984) 157–161.

[13] P.G. Selinger, M.M. Astrahan, D.D. Chamberlin, R.A. Lorie and T.G. Price, Access path selection in a relational database system, *Proc. ACM SIGMOD* (Boston, 1979) 23–34.

[14] B.T. Vander Zanden, H.M. Taylor and D. Bitton, A general framework for computing block accesses, *Informat. Syst.* 12(2) (1986) 177–190.

[15] S.B. Yao, Approximating block accesses in database organizations, *Comm. ACM* 20(4) (Apr. 1977) 260–261.

[16] G.K. Zipf, *Human behaviour and the principle of least effort* (Addison-Wesley, Reading, 1949).



Fabio Grandi received the Laurea in Electronics Engineering from the University of Bologna, Italy, in 1988.

Since 1989 he has worked at the C.I.O.C. center of the Italian National Research Council (CNR) in Bologna, supported by a fellowship from the CNR, in the field of neural networks and temporal databases. He finished his Ph.D. course at the Department of Electronics, Computer Science and Systems (DEIS) of the University of Bologna in 1993. In 1993 and 1994 he was an Adjunct Professor at the Universities of Ferrara, Italy, and Bologna.

He is currently with the DEIS as a Research Associate. His scientific interests include temporal databases, storage and access structures, access cost models and information retrieval systems. He is a member of the TSQL2 language design committee.

Maria Rita Scalas received the Laurea in Physics from the University of Bologna, Italy, in 1974.

From 1975 to 1979 she worked at the Universities of Pisa and Bologna supported by a four-year fellowship from the Italian Ministry of Education. In 1980 she became a Research Associate in Computer Science at the University of Bologna and a consultant at the C.I.O.C.-CNR center of the National Research Council in Bologna. In 1986 she was a visiting scientist at the IBM Scientific Center in Heidelberg, Germany, where she took part in the AIM-P project. In 1987 she became an Associate Professor at the University of Trieste, Italy.

She is presently with the Department of Electronics, Computer Science and Systems, University of Bologna. Her research interests are in the area of database management systems, temporal databases, access structures, optimizers and database design.