BIT 30 (1990), 620–630

ANALYSIS FOR THE END OF BLOCK WASTED SPACE

YANNIS MANOLOPOULOS and CHRISTOS FALOUTSOS*

Department of Electrical Engineering
Aristotelian University of Thessaloniki,
54006 Thessaloniki, Greece.

Department of Computer Science,
University of Maryland,
College Park, MD 20742, USA.

Abstract.

The problem examined in this report is the calculation of the average wasted space at the end of the block when variable length records are inserted in the file. Previous efforts are based in approximations. Here, a detailed analysis based on Markov chains gives the exact solution. A framework is presented which shows the relations between the previous approaches. The proposed model includes the previous models as special limiting cases. Simulation results close to the analytic results are also presented.

CR categories and subject descriptors: D.4.2, D.4.8, H.2.2.

General terms: Experimentation, Performance.

Key-words: End of Block Wasted Space, Variable Length Records.

1. Introduction.

Precise estimations of the disk storage requirements are important for database designers and practitioners [5]. Simple notion rules for the required space are needed in query optimizers [11], block selectivities [1, 2] and related physical database design problems. The problem examined in this report is to calculate the average wasted space at the end of the block when variable length records are inserted in the file. Variable length records are very often met in database environments due to a number of reasons, such as variable length fields, missing or multiple attribute values and compression [3, 9, 10]. Previous efforts are based on approximations. We present a detailed analysis giving the exact solution and a framework showing the relations between the previous approaches. Simulation results are also reported.

The structure of the paper is as follows. Section 2 describes and compares

* This research was sponsored partially by the National Science Foundation under the grants DCR-86-16833, IRI-8719458 and IRI-8958546 and by the Air Force Office of Scientific Research under grant AFOSR-89-0303.

Received October 1989. Revised May 1990.

previous efforts and section 3 contains the analysis. Section 4 discusses the results and gives some arithmetic examples for the case of two record types.

2. Survey-problem description.

In the sequel it is assumed that (a) blocks are the I/O units, (b) records are stored in blocks in a predescribed manner, for example according to key order or sequentially, and (c) record spanning in two blocks is not permitted.

The third assumption indicates that whenever a record does not fit at the end of a specific block, then it is written in the next block leaving the previous one not completely full. Evidently, more sophisticated software may make better use of the disk storage according to some fitting algorithm. This method, however, would demand additional space and time cost for the storage and maintenance of a table with the relevant information. The second assumption suggests that future records do not use the previous block and will therefore waste some space. In this way no record is fetched with two block accesses but faster secondary access results in low storage utilization. Note that current systems use preformatted disks, therefore blocks have constant size. The problem arising is to estimate the size of the wasted space and determine under what circumstances it is really considerable.

Suppose that consecutive blocks of 500 bytes each are given. Incoming records belong to two types of 300 and 200 bytes respectively, with equal probabilities. Previous works on the subject reported approximate analyses estimating the wasted space. For the data of this example they estimate the wasted space as depicted in Table I.

Table 1. *Example on previous results for the wasted space problem.*

Hakola-Heiskanen (1980)	130 bytes of wasted space
Hubbard (1981)	129.5 bytes
Teorey-Fry (1982)	75 bytes
Wiederhold (1983)	125 bytes

Hand experimentation gives the exact results of Table II. The two record types, although arriving with equal probabilities, are not actually stored with same ones. The storage probabilities are denoted PS_1 and PS_2; they vary from block to block.

Table 2. *Transition of statistics for the first 3 blocks of the example.*

	PS_1	PS_2	NR	RL	WS
1st block	42.86%	57.14%	1.75	242.86	75.
2nd block	48.15%	51.85%	1.69	248.15	81.25
3rd block	49.53%	50.47%	1.67	249.53	82.81

As a consequence, the same is true for the mean values of (a) the total number of records per block (NR), (b) the record length per block (RL) and (c) the wasted space per block (WS).

Let us examine a little closer the results of the references in order of appearance. If we denote the wasted block size when the block size is BS by $WS(BS)$, then Wiederhold's approximation [13] for the wasted space is equal to half the mean record length, i.e.,

$$WS(BS) = \mu/2 \tag{1}$$

where:

$$\mu = E[\mathrm{Li}] = \sum_{i=1}^{t} L_i P_i$$

where L_i is the record length and P_i the corresponding probability for the record of type i ($1 \leq i \leq t$). The number of record types is denoted by t.

Hakola and Heiskanen [6] use renewal theory [4,8] and conclude that the wasted space is greater than Wiederhold's approximation. More specifically they do a second order approximation ending up with the formula:

$$WS(BS) = \mu/2 + \sigma^2/2\mu \tag{2}$$

where σ is the standard deviation of the record length distribution:

$$\sigma^2 = E[(L_i - \mu)^2]. \tag{3}$$

Note that the first term is exactly Wiederhold's approximation.

Hubbard [7] follows the same line of thought as Hakola and Heiskanen, with more intuitive probabilistic argument, resulting in the following formula:

$$WS(BS) = E[L_i^2]/(2\mu) - 1/2$$

which is equivalent to:

$$WS(BS) = \mu/2 + \sigma^2/2\mu - 1/2.$$

This is exactly the formula of Hakola and Heiskanen, except for the term 1/2; the difference is due to the fact that they implicitly assume a continuous, while Hubbard assumes a discrete record length distribution. The common assumption of the approaches above is that the block size is very large compared to the record sizes.

Teorey and Fry [12] outline an exact way to calculate the average wasted block space for the very first block. They enumerate all the possible combinations of the multiple-type records that fit in the first block, calculate the wasted space in each case, and average over all combinations. They observe that their result will be an approximation for the subsequent blocks, the reason being that the subsequent blocks have to accomodate the overflow records from their predecessor; this fact changes the distribution of record lengths that arrive in each block for storage.

Next a model and the exact analysis for the average wasted space per block is presented. The first step is to calculate the probability distribution of the length of

the first record stored in the jth block. By-products of this analysis are: (a) the mean value of records in the jth block, (b) the mean record length of the records in the jth block, and (c) the record type distribution of the records stored in the jth block.

3. Analysis.

The parameters of the problem are summarized in Table III. Let BS be the number of bytes available per any block, ignoring the space occupied by the header block, identifiers and pointers. The population of records is divided into t types with arriving probabilities P_i and lengths L_i, $1 \leq i \leq t$. Let P_i be known in advance and $L_i < L_{i+1}$ without loss of generality. Record lengths are assumed to be independent of the primary key values. The assumption of a fixed number of types is justified from the fact that frequently variable length records are the result of missing attribute values for certain attributes, or repetitions of a certain set of attribute values several times, or mixing several record types with a common key. In environments where variable length records are the result of some compression scheme and a continuous probability distribution of record lengths is observed, the range of lengths can be subdivided into subranges and a type can be identified with a subrange.

Table 3. *List of parameters.*

BS	Block size in bytes
t	Number of record types
L_i	Type i record length in bytes ($1 \leq i \leq t$)
P_i	Probability of type i arriving records
NR_{ij}	Number of type i records in the jth block
NR_j	Total number of records in the jth block
PS_{ij}	Probability a type i record is stored in the jth block
RL_j	Mean record length in the jth block
$P_{ij}(BS)$	Probability a type i record intercepts the boundary of the jth block of size BS given a specific record selection
$PIN_{ij}(BS)$	Probability a type i record intercepts the boundary of the jth block of size BS
$OC_j(BS)$	Occupied space in bytes in the jth block of size BS
$WS_j(BS)$	Wasted space in bytes in the jth block of size BS

The problem is defined as follows:

GIVEN: the record lengths L_i, the record arriving probabilities P_i and the block size BS,

FIND: the average wasted space $WS_j(BS)$ for the jth block.

Suppose NR_{ij} is the number of records of type i in the jth block and NR_j is the total number of records in the jth block:

$$NR_j = \sum_{i=1}^{t} NR_{ij}. \tag{4}$$

For efficiency reasons, the number of cases may be limited observing that NR_j's are bounded by $x_2 = \lfloor BS/L_1 \rfloor$ and $x_1 = \lfloor BS/L_t \rfloor$.

Suppose now that the file is being loaded and only the first block is under consideration. Let x be the number of records that have already arrived. The probability that NR_{11} of these records have been selected from type 1, NR_{21} from type 2, ..., NR_{t1} from type t is:

$$q(NR_{11}, \ldots, NR_{t1}) = \left(\sum_{k=1}^{t} NR_{k1} \right)! \prod_{k=1}^{t} P_k^{NR_{k1}} / \prod_{k=1}^{t} NR_{k1}! \tag{5}$$

The sum of the lengths of these records is $\sum_{i=1}^{t} NR_{i1} L_i$. This quantity may exceed the size of the first block. Let NR_1 out of x be the number of records which are stored in the first block. (NR's are random variables.) Let $Q(NR_1)$ be the probability that exactly NR_1 records are stored within the first block. Then $Q(NR_1)$ is:

$$Q(NR_1) = \sum_{NR_1} q(NR_{11}, \ldots, NR_{t1}) \cdot \sum_{j} P_j \tag{6}$$

under the condition (4) and subject to additional constraints for every j:

$$\sum_{i=1}^{t} NR_{i1} L_i \leq BS < \sum_{i=1}^{t} NR_{i1} L_i + L_j. \tag{7}$$

This formula is explained as follows. Remember that the record lengths are independent of the key values. Therefore in finding $Q(NR_1)$ it can be considered that the NR_1 records are randomly selected one after another from the underlying population of variable length records. The order of selection cannot change. Thus if the next selection involves a long record which will result in an overflow, the empty space in the block cannot be covered by a subsequently selected short record. The first summation above involves all the selections of NR_1 records from the t types such that the sum of the lengths of the records is less or equal to the block size. Such a selection has probability $q(NR_1, \ldots, NR_t)$ as given by (5). The conditions above guarantee that the NR_1 records in the first block (and no more) the $(NR_1 + 1)$th record must have a length greater than the empty space left in the block. The probability that this happens is calculated in the right hand summation.

The probability distribution of the length of records in the successive blocks is not the same as the probability distribution of the record lengths in the first block. The reason is that longer records have higher probability than shorter ones to be intercepted by the first block boundary. Thus longer records are more likely to be found in the beginning of every block starting from the second and thereafter. Let $P_{i1}(BS)$ be the total probability of any arrangement of records in the first block of size BS so that a record of type i is intercepted by a block boundary. Then:

$$P_{i1}(BS) = \sum_{NR_1 = x_1}^{x_2} q(NR_{11}, \ldots, NR_{t1}) P_i \tag{8}$$

where every i satisfies the following relation:

$$\sum_{j=1}^{t} NR_{j1} L_j \leq BS < \sum_{j=1}^{t} NR_{j1} L_j + L_i.$$

This formula is derived in a similar manner as formula (7) taking into account that the record with the key value next in order may not fit within the block and therefore may have to move to the next block.

The probability that a record intercepted by a block boundary is of type i is:

$$PIN_{i1}(BS) = P_{i1}(BS) / \sum_{i=1}^{t} P_{i1}(BS). \tag{9}$$

Now it is easy to calculate the wasted space at the end of the first block. Under the conditions (4) and (7) the occupied space is:

$$OC_1(BS) = \sum_{NR_1 = x_1}^{x_2} q(NR_{11}, \ldots, NR_{t1}) \cdot \sum_i NR_i L_i P_j \tag{10}$$

and the wasted space is:

$$WS_1(BS) = BS - OC_1(BS). \tag{11}$$

The mean number of records of type i in the first block is given by the following formula:

$$\overline{NR}_{i1}(BS) = \frac{\sum_{NR_1 = x_1}^{x_2} \sum_{NR_{i1}=1}^{NR_1} q(NR_{11}, \ldots, NR_{t1})\, NR_{i1}\, P_j}{\sum_{k=1}^{t} \sum_{NR_1 = x_1}^{x_2} \sum_{NR_{i1}=1}^{NR_1} q(NR_{11}, \ldots, NR_{t1})\, NR_{k1}\, P_j}. \tag{12}$$

The mean total number of records in the first block is given by the following formula:

$$\overline{NR}_1(BS) = \sum_{i=x_1}^{x_2} iQ(i). \tag{13}$$

The probability distribution of the lengths of the records which are actually stored in the first block is:

$$PS_{i1} = \overline{NR}_{i1}(BS) / \overline{NR}_1(BS). \tag{14}$$

The mean record length in the first block is equal to:

$$RL_1 = OC_1 / \overline{NR}_1. \tag{15}$$

Consider now the second block. The probability $PIN_{12}(BS)$ is the following sum of products:

$$PIN_{12}(BS) = \sum_{j=1}^{t} PIN_{11}(BS)\, PIN_{11}(BS - L_j). \tag{16}$$

In an analogous manner the probability distribution of the intercepted records at the nth block is equal to:

$$PIN_{in}(BS) = \sum_{j=1}^{t} PIN_{i,n-1}(BS)\, PIN_{i1}(BS - L_j). \tag{17}$$

Note that the process forms a Markov chain because the statistics of the nth block depend only on the state of the $(n-1)$th block.

Therefore, it is easy again to calculate the wasted space for the second block and thereafter. That is:

$$WS_2(BS) = \sum_{i=1}^{t} PIN_{i1}(PS)\, WS_1(BS - L_i) \tag{18}$$

and in general:

$$WS_n(BS) = \sum_{i=1}^{t} PIN_{i,n-1}(BS)\, WS_1(BS - L_i). \tag{19}$$

The following measures can also be derived. The mean number of records in the nth block is:

$$\overline{NR}_n(BS) = \sum_{i=1}^{t} PIN_{i,n-1}(BS)\, \overline{NR}_1(BS - L_i) + 1 \tag{20}$$

and the mean record length in the nth block is:

$$RL_n = (BS - WS_n(BS))/\overline{NR}_n. \tag{21}$$

The probability distribution of the lengths of the records of type j which are actually stored in the nth block is:

$$PS_{jn} = \overline{NR}_{jn}(BS)/\overline{NR}_n(BS) \tag{22}$$

where

$$\overline{NR}_{jn}(BS) = \sum_{i=1}^{t} PIN_{i,n-1}(BS)\, \overline{NR}_{j1}(BS - L_i) + PIN_{j,n-1}(BS). \tag{23}$$

4. Results and discussion.

Previous efforts fall in two categories. The first one assumes that the block size is very large. Under this assumption the statistical characteristics of all the blocks are identical. The reason is that the intercepted record by the boundary of the jth block is much shorter than the block to have any significant effect. Thus:

$$PIN_{in} = PIN_{i1} \quad \text{for every } n.$$

Formula (10) gives the exact way to calculate PIN_{in} using the polynomial distribution. This is in direct agreement with Hakola et al. who suggested a renewal theory

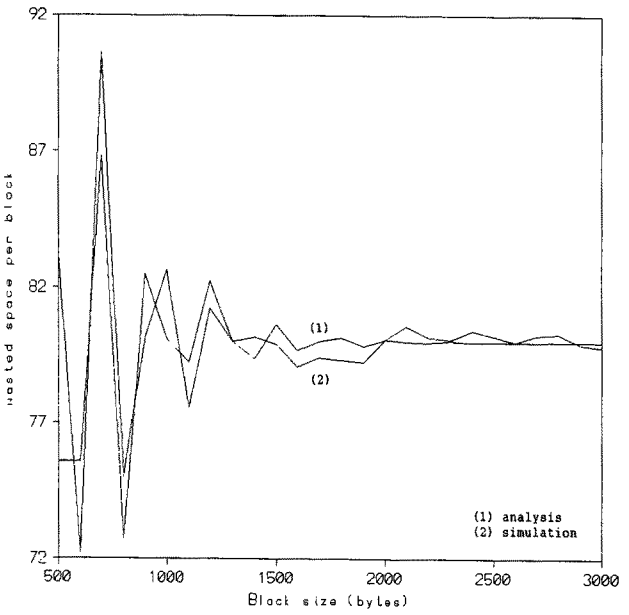

Fig. 1. Wasted space per block as a function of the block size (in bytes) by analysis and simulation. $L_1 = 300$, $L_2 = 200$ and $P_1 = P_2 = 0.5$.

based approach to approximate the statistical characteristics of interest.

In the second category Teorey and Fry calculate the wasted space only for the first block. Our formula (11) is the mathematical expression of this. Our work goes further (equation 19) and gives the statistics of the subsequent blocks.

Extensive experiments with arithmetic examples for two record types lead to the following observations. The two curves of Figure 1, produced by using our analytic formula and by simulation, illustrate the wasted space as a function of the block size at the steady state. The two record types are $L_1 = 300$ and $L_2 = 200$ bytes while the arriving probabilities are $P_1 = P_2 = 0.5$, values which may appear in real life problems. Note that simulation results are very close to those produced by the analysis. The corresponding results produced by previous efforts are also given in Table 1.

Figure 2 is produced by assuming the two record types of $L_1 = 300$ and $L_2 = 200$ bytes, while the probabilities P_1 and P_2 vary from 0 to 1 for every type. Curves (1), (2) and (3), which correspond to the previous research efforts, hold for any block size. Curves (4) and (5) correspond to the case that the block size is equal to 1000 and 2400 bytes respectively. Note, here, that 2400 bytes is the block size of the IBM 3380 disk system, and that the endpoints of all curves may be calculated very easily.

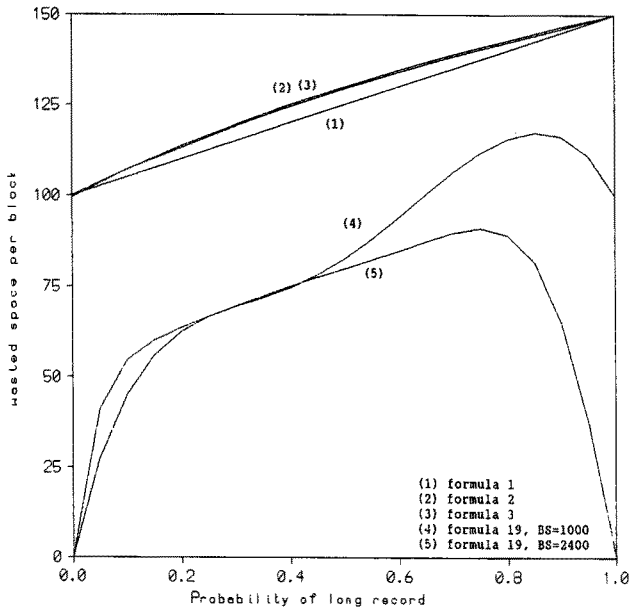

Fig. 2. Wasted space per block as a function of the arriving probability of the long record. $L_1 = 300$, $L_2 = 200$.

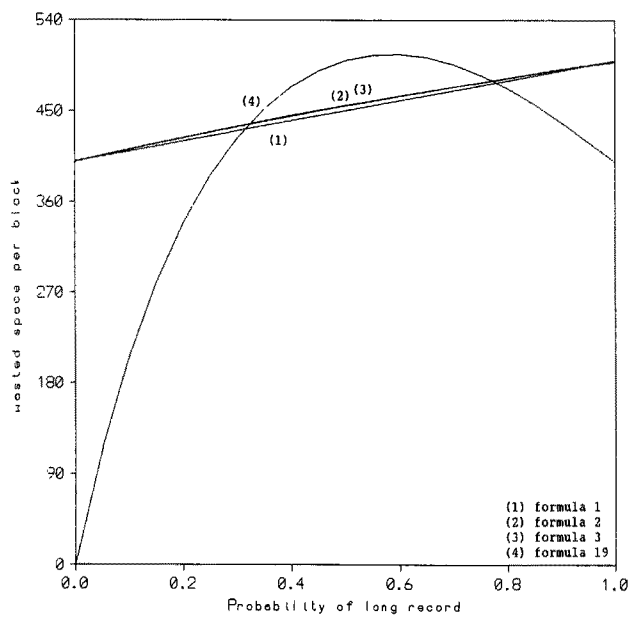

Fig. 3. Wasted space per block of the arriving probability of the long record. $L_1 = 1000$. $L_2 = 800$ and $BS = 2400$.

Figure 3 illustrates the wasted space for the two record types of $L_1 = 1000$ and $L_2 = 800$ bytes, while the probabilities P_1 and P_2 vary from 0 to 1 for every type. The block size is again 2400 bytes.

Note that the curve produced by our analysis in some cases may give values greater than the values of previous analysis.

Table 2. *Transition of statistics for the first 3 blocks of the example.* $BS = 2400$, $L_1 = 200$, $L_2 = 300$

	$PS_{1j} = 0.5$ $P_2 = 0.5$			$P_1 = 0.65$ $P_2 = 0.35$			$P_1 = 0.35$ $P_2 = 0.65$		
	PS_{1j}	RL	WS	PS_{1j}	RL	WS	PS_{1j}	RL	WS
$j = 1$	51.07	248.93	79.98	65.97	234.03	72.34	35.99	264.01	87.35
$j = 2$	50.00	250.00	79.98	65.00	235.00	72.34	35.00	265.00	87.43
$j = 3$	50.00	250.00	79.98	65.00	235.00	72.34	35.00	265.00	87.43

The steady state is reached very quickly, at most after $j = 4$ or 5 blocks. For example, Table 4 depicts the quick stabilization of the statistics for the first few blocks for some specific cases of parameter values. From the same table and numerous other examples we see that in the steady state the records are stored with probabilities equal to the arriving ones. Therefore, the average length of the stored records in the jth block ($j > 4$) is equal to the average record length of the arriving records. In cases where record sizes and block size are (almost) multiple of each other, then the infinite block size approach leads to very pessimistic results. For example, in Table 5 the effect on the wasted space is depicted for some specific extreme cases. In the simple case when there is only one record type and the block size is a multiple of the record size, then the previous efforts lead to erroneous results.

Table 5. *Wasted space for record and block sizes (almost) multiples of each other.* $BS = 2400$.

Formula	$L_1 = 400$ $P_1 = 0.5$ $L_2 = 400$ $P_2 = 0.5$	$L_1 = 1200$ $P_1 = 0.95$ $L_2 = 600$ $P_2 = 0.05$	$L_1 = 1200$ $P_1 = 0.95$ $L_2 = 1100$ $P_2 = 0.05$
(1) Wiederhold	200	585	597.5
(2) Hakola-Heiskanen	200	592.31	597.7
(3) Hubbard	199.5	591.81	597.2
(19)	0	51.99	10

In conclusion, the contributions of this work are the following. The wasted space problem is formulated as a Markov chain and the exact solution closes the case. Also, simulation results provided confer the analysis. Any combination of physical parameters may be handled, while previous methods deviate greatly from the exact solution under "maliciously" chosen parameters. Our analytic formula has $O(t)$

complexity, therefore for small values of t it may be used by practitioners for estimating disk storage requirements. Future research should examine the case of more than two record types.

Acknowledgement.

Thanks are due to the anonymous referee and the editor who helped in improving the presentation of the work.

REFERENCES

[1] S. Christodoulakis, *Estimating block selectivities*, Information Systems, Vol. 9, No. 1, 1984.
[2] S. Christodoulakis, *Implications of certain assumptions in database performance evaluation*, ACM Transactions on Database Systems, Vol. 9, No. 2, pp. 163–187, 1984.
[3] S. Christodoulakis, Y. Manolopoulos and P. Å. Larson, *Analysis of overflow handling for variable length records*, Information Systems, Vol. 14, No. 1, pp. 151–162, 1989.
[4] D. R. Cox, *Renewal Theory*, Methuen, London, 1967.
[5] C. J. Date, *Introduction to Database Systems*, John Wiley, Vol. 1, 4th edition, 1988.
[6] J. Hakola and A. Heiskanen, *On the distribution of wasted space at the end of file blocks*, BIT, Vol. 20, No. 2, pp. 145–156, 1980.
[7] G. U. Hubbard, *Computer-assisted Database Design*, Van Nostrand Reinhold Company, 1981.
[8] L. Kleinrock: *Queueing Systems*, Vol. I: Theory, John Wiley, NY, 1975.
[9] Y. Manolopoulos and S. Christodoulakis, *File organizations with shared overflow blocks for variable length objects*, submitted.
[10] Y. Manolopoulos and N. Fistas, *Algorithms for a hash based file with variable length records*, Information Sciences, to appear.
[11] T. Sellis, *Global query optimization*, ACM Transactions on Database Systems, Vol. 13, No. 1, pp. 23–53, 1988.
[12] T. J. Teorey and J. P. Fry, *Design of Database Structures*, Prentice Hall, NJ, 1982.
[13] G. Wiederhold, *File Organization for Database Design*, McGraw-Hill, NY, 1987.