Optimal File Distribution For Partial Match Retrieval

Kim, Myoung Ho
&
Pramanik, Sakti

Michigan State University
Computer Science Department,
East Lansing,MI 48824-1027

Abstract

In this paper we present data distribution methods for parallel processing environment. The primary objective is to process partial match retrieval type queries for parallel devices.

The main contribution of this paper is the development of a new approach called FX (Fieldwise eXclusive) distribution for maximizing data access concurrency. An algebraic property of exclusive-or operation, and field transformation techniques are fundamental to this data distribution techniques. We have shown through theorems and corollaries that this FX distribution approach performs better than other methods proposed earlier. We have also shown, by computing probability of optimal distribution and query response time, that FX distribution gives better performance than others over a large class of partial match queries. This approach presents a new basis in which optimal data distribution for more general type of queries can be formulated.

1. Introduction

Parallel processing in databases is important because it can maximize concurrency in query processing. As a generalized model for parallel processing system in databases, two stage parallel processing can be used. These are data distribution and data construction stages [PrKi88]. Data distribution stage determines how the data can be distributed to parallel processing nodes so that maximum concurrency is achieved between the processing nodes. Data construction stage, on the other hand, builds the appropriate structure of the local data, suitable for accessing by the local processing nodes. These two stages can be dependent, and the local data structure can be distributed among the devices. The proposed parallel processing model has been used in multi-directory hashing [PrDa86], and HCB_tree [PrKi87]. Other approaches to parallel processing of database systems have also been proposed in the past [LeRo85, Pram86, RoJa87, SNEL79].

In this paper we present data distribution methods for partial match retrieval type queries. The strategies for data construction within local devices is not discussed here.

Partial match queries are queries where some of the attributes are specified, hence a set of qualified records need to be retrieved. When a file is constructed on parallel devices, it is important to store these records to maximize concurrency. This also helps in balancing the work load in each device.

It is believed that multi-key hashing is effective for partial match retrieval type applications. Multi-key hash function, H, for a database consisting of n fields is a set of n functions $\{H_1, \cdots, H_n\}$ such that given a record $r = <r_1, \cdots, r_n>$, $H(r) = <H_1(r_1), \cdots, H_n(r_n)>$. $H(r)$ is usually called a bucket. Rivest [Rive76] and Rothnie, et al. [RoLo74] have independently proposed the use of multi-key hashing, as an alternative to inverted files, to reduce the total search time for partial match retrieval type queries. In [RoLo74] it has been shown that multi-key hashing scheme is superior to inverted file techniques. They have also proposed hybrid approach which combines the above two techniques. The design of multi-key hash functions was considered in [Burk76, Burk79]. The determination of each field size for minimum search time based on query statistics was also investigated by [RoLo74, Bolo79, AhUl79]. In [Du85] it has been shown that the problem of finding the optimal field sizes for multi-key hashing scheme is NP-hard. The main focus of those research is on minimizing the total number of bucket accesses. In this research our objective is to achieve maximum parallelism by distributing buckets in multi-key hashing.

The data distribution is said to be optimal for a partial match query, when no device has more than $\lceil$total number of qualified buckets / number of devices$\rceil$ buckets. It has been shown in [Sung87] that there does not exist an optimal data distribution method in certain types of file systems. However, the tight bound of sufficient and necessary conditions for the existence of optimal data distribution in any file system has not been found.

There are a few heuristic methods for distributing data in partial match retrieval type queries. Du, et al. have proposed data distribution method based on modulo allocation [DuSo82]. Modulo allocation is simple but does not work in many cases. For example, it may not give optimal distribution if some of the field sizes are less than the given number of devices. So, for a large number of parallel processing nodes such as Butterfly machines[BBN86, CGSTMB85], Modulo distribution may not be appropriate. GDM (Generalized Disk Modulo) method has also been proposed in [DuSo82] to overcome this problem. This method gives a sufficient condition to achieve optimal distribution. However, no general method has been given to find the optimal distribution parameters. In fact, the problem of finding the optimal parameter values could be very complex [DuSo82]. Since modulo distribution does not work well for

This work was partially supported by a grant from the Naval Research Laboratory under Contract #N00014-87-K-2022

binary cartesian product file (binary cartesian product file is a cartesian product file [CLD80, DuSo82] in which each attribute contains only two elements), other heuristics have been proposed by [Du82, Sung85]. These heuristics are also special cases of GDM. Several useful properties of these modulo based distribution methods have also been given in [Sung87]. Data distribution methods based on minimal spanning trees and short spanning paths have also been proposed in [FaRC86].

In this paper we propose FX(Fieldwise eXclusive-or) distribution method which gives better performance for a wider range of parameter values than existing methods. The main idea behind the FX distribution method is the use of bitwise exclusive-or operation on the field values which are computed by multi-key hashing. Here, we show several useful characteristics of exclusive-or operation for optimal data distribution. Field transformation techniques have been used to extend the scope of optimalities in FX distribution.

The remainder of this paper is organized as follows. In section 2, we describe notations and terminology. In section 3, we define Basic FX distribution method and its optimality conditions. In section 4, we describe field transformation techniques for fields whose sizes are less than the number of devices. The extended optimality conditions for these field transformation schemes are also identified. In section 5, we compare the performance of FX distribution method with those of the others. Section 6 contains concluding remarks.

2. Notation and Terminology

Before describing FX distribution method, it is necessary to introduce some notations as well as relevant definitions and assumptions.

Definition :

- $f_i = \{0, 1, \ldots, F_i-1\}$, a set of hashed values of field i.
- F_i denotes $|f_i|$.
- M denotes the number of parallel devices.
- N is the set of all natural numbers including 0.
- Z_M is the set of all integers from 0 to M-1.
- $(a_{m-1} \ldots a_0)_B$ is a binary notation of an integer, where a_i is a binary digit.

$|f_i|$ is assumed to be a power of 2 which is common for hash directory files for partitioned [AhUl79] or dynamic hashing schemes [FJNH79, Lars78, Lars80, Litw80]. The number of devices M is also assumed to be a power of 2.

Definition : Let $f_1 \times f_2 \times \ldots \times f_n$ be a set of all buckets. When the given number of devices is M, data distribution method is a function FD : $f_1 \times f_2 \times \ldots \times f_n \rightarrow Z_M$.

Definition : Let R(q) be the set of buckets which satisfy qualifications for a partial match query q. The distribution method is called *strict optimal* for a partial match query in a given file system if each device has no more than $\lceil |R(q)|/M \rceil$ number of buckets. When the distribution method is strict optimal for all possible partial match queries in a given file system, it is called *perfect optimal* for that file system.

Definition : The distribution method is called k-optimal, $0 \le k \le n$, for a given file system consisting of n fields, when it is strict optimal for all partial match queries which have exactly k unspecified fields.

Thus, the distribution method is perfect optimal, if it is k-optimal for all k = 0, ... , n. Note that some authors exclude cases where the number of unspecified fields is 0 (i.e. exact match) and the number of unspecified fields is n (i.e. retrieval of whole file) from partial match queries.

Definition : [+] denotes exclusive-or operation between two bits. We will use the same notation [+] to denote exclusive-or operation between integers and sets of integers as follows. When $X = (a_{m-1} \ldots a_0)_B$ and $Y = (b_{m-1} \ldots b_0)_B$ are two integers, $X [+] Y = (a_{m-1} [+] b_{m-1} \ldots a_0 [+] b_0)_B$. If X is an integer and $Y = \{y_1, \ldots, y_L\}$ is a set of integers, X [+] Y is defined as $\{X [+] y_i \mid y_i \in Y\}$. If both $X = \{x_1, \ldots, x_K\}$ and $Y = \{y_1, \ldots, y_L\}$ are sets of integers, X [+] Y is defined as $\{x_i [+] y_j \mid x_i \in X, y_j \in Y\}$

For example, if $X_1 = 2$ and $Y_1 = 3$ then $X_1 [+] Y_1 = 1$. If $X_2 = 2$ and $Y_2 = \{0, 1, 2, 3\}$ then $X_2 [+] Y_2 = \{0, 1, 2, 3\}$.

Definition : $\overset{n}{\underset{i=1}{[+]}}(Y_i) = Y_1 [+] Y_2 [+] Y_3 \cdots [+] Y_n$.

Note that [+] operator is associative and $\overset{n}{\underset{i=1}{[+]}}$ is a shorthand notation for exclusive-or operation between sets of integers $Y_1, Y_2, \ldots, Y_n$.

3. Basic FX Distribution

Let $f_1 \times f_2 \times \ldots \times f_n$ be a set of all buckets. Basic FX distribution method allocates bucket $<J_1, \ldots, J_n>$ into device $T_M \left[\overset{n}{\underset{j=1}{[+]}}(J_j)\right]$, where T_M : $N \rightarrow Z_M$ is a function which returns only the rightmost $\log_2 M$ bits of domain values, and $J_j \in f_j$ for j = 1, ... , n.

Example 1. Table 1 shows the bucket distribution by Basic FX distribution method, where $f_1 = \{0, 1\}$, $f_2 = \{0, 1, 2, 3, 4, 5, 6, 7\}$ and M = 4. In this table, binary numbers are used for field values and decimal numbers are used for Device No. (This convention will be used in all examples of FX distribution). Here, Device No = $T_M \left[J_1 [+] J_2\right]$, where $J_1 \in f_1, J_2 \in f_2$ and T_M returns the rightmost two bits of the result of $J_1 [+] J_2$.

f_1	f_2	Device No
000	000	0
000	001	1
000	010	2
000	011	3
000	100	0
000	101	1
000	110	2
000	111	3
001	000	1
001	001	0
001	010	3
001	011	2
001	100	1
001	101	0
001	110	3
001	111	2

Table 1. Basic FX distribution

As shown in Table 1, Basic FX distribution is strict optimal for any partial match query in a file system of example 1. For example, if first field value is $(001)_B$ and second field value is unspecified, then we have to access eight buckets $<(001)_B,(000)_B>$, ... , $<(001)_B,(111)_B>$. Since each device has two qualified buckets for this partial match query, FX distribution is strict optimal for this query.

Lemma 1.1. Z_M is a set which contains M different nonnegative integers from 0 to M-1. Let k be some integer $0 \le k \le M-1$. Then $Z_M [+] k = Z_M$.
<proof> Given in [KiPr87]

Example 2. Let $Z_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$ and k = 3. Then $Z_8 [+] k = \{3, 2, 1, 0, 7, 6, 5, 4\} = Z_8$.

Theorem 1. Basic FX distribution is always 0-optimal and 1-optimal.
<proof> (1) 0-optimal : This is trivially true.
(2) 1-optimal : Let only one field *i* be unspecified and $T_M \left[\overset{}{\underset{j \neq i}{[+]}}(J_j)\right] = h$, where J_j is the specified value of field j. Thus, h gives the projection of

the rightmost $\log_2 M$ bits of the value obtained by doing the exclusive-or of all the specified values of the query. There are two cases, $F_i \le M$ and $F_i > M$. When $F_i \le$ M, for all I $\in f_i$, $T_M(I)$ [+] h is different from each other (This is a direct consequence of Lemma 1.1.) Therefore, the distribution is optimal. (Note that $T_M(A[+]B) = T_M(A)[+]T_M(B) = T_M(A[+]T_M(B)) = T_M(T_M(A)[+]T_M(B)$. This is true because the bits whose positions are higher than or equal to $\log_2 M$ do not affect the final result.) When $F_i >$ M, let F_i = A*M. By Lemma 1.1 f_i [+] h = Z_{A*M}. Since $\#\{\alpha \in Z_{A*M} \mid T_M(\alpha) = z\}$ = A for any $z \in Z_M$, FX distribution allocate A number of buckets to each device. (Note that # denotes the cardinality of a set.) Therefore, the distribution is optimal. □

Theorem 1 says that Basic FX distribution is strict optimal for any partial match query in which the number of unspecified fields is 0 or 1. Note that in the above expression, $\#\{\alpha \in Z_{A*M} \mid T_M(\alpha) = z\}$ = A, A denotes the number of qualified buckets that correspond to a particular device z.

Theorem 2. For any partial match query which has two or more unspecified fields, Basic FX distribution is strict optimal, if there exists at least one unspecified field i such that $F_i \ge M$.

<proof> For partial match query q, let q(f) = $\{i_1, i_2, \ldots, i_k\}$ be the set of fields which are unspecified and the size of at least one of these fields is greater than or equal to M. Without loss of generality, let us assume that $F_{i_1} \ge M$. Thus, $F_{i_1} = A_{i_1}*M$, $A_{i_1} \in$ N, $A_{i_1} \ge 1$. Let $h = T_M\left[\underset{j \in q(f)}{[+]}(J_j)\right]$, where J_j denotes the specified value of field j. By Lemma 1.1, h [+] $f_{i_1} = Z_{A_{i_1}*M}$ and hence $\#\{J_{i_1} \in f_{i_1} \mid T_M(h[+]J_{i_1}) = z\} = A_{i_1}$ for all $z \in Z_M$. Here, we have done exclusive-or of h with the set of values of the unspecified field i_1. A_{i_1} gives the number of unique values of the unspecified field i_1 that correspond to a particular value z. Now we will exclusive-or the set of values h [+] f_{i_1} with a value of the unspecified field i_2. Let $J_{i_2} \in f_{i_2}$. By Lemma 1.1 $\#\{J_{i_1} \in f_{i_1} \mid T_M(h[+]J_{i_1}[+]J_{i_2}) = z\} = A_{i_1}$ for all $z \in Z_M$. Note that this will not change the number of unique values of field i_1 that correspond to a particular z due to Lemma 1.1. (We also used the equality, $T_M(h[+]J_{i_1}[+]J_{i_2}) = T_M(h[+]J_{i_1}[+]T_M(J_{i_2}))$.) Thus, $\#\{(J_{i_1}, J_{i_2}) \in f_{i_1} \times f_{i_2} \mid T_M(h[+]J_{i_1}[+]J_{i_2}) = z\} = A_{i_1}F_{i_2}$ for all $z \in Z_M$. Here, the number of unique values for each z is more by a factor F_{i_2} because the size of the domain has increased by a factor F_{i_2}. By continuing this argument, $\#\{(J_{i_1}, \cdots, J_{i_k}) \in f_{i_1} \times \cdots \times f_{i_k} \mid T_M\left(h \underset{p=1}{\overset{k}{[+]}}(J_{i_p})\right) = z\} = A_{i_1} \prod_{p=2}^{k} F_{i_p} = \left(\prod_{p=1}^{k} F_{i_p}\right) / M$ for all $z \in Z_M$. □

Note that Theorem 1 works for partial match queries with only one unspecified field while Theorem 2 applies to partial match queries with more than one unspecified fields.

Theorem 1 and 2 show general characteristics of exclusive-or operation for optimal file distribution. This is mainly due to the property described in Lemma 1.1. However, Basic FX distribution does not give optimal distribution for partial match queries with 2 or more unspecified fields, when the size of none of the unspecified fields is greater than or equal to M. For example, when M = 16 and all others are the same as in example 1, the distribution is not optimal. Theorem 3 gives the sufficient conditions for optimal distribution for these cases.

Theorem 3. Let q(f) = $\{i_1, i_2, \ldots, i_k\}$ be the set of unspecified fields for partial match query q, where $F_j <$ M, for all j $\in$ q(f). FX distribution is strict optimal for partial match query q, if there exist a set of fields $\{i_1, \ldots, i_j\} \subseteq$ q(f) such that $|f_{i_1} \times \cdots \times f_{i_j}| \ge M$ and $\#\{(J_{i_1}, \cdots J_{i_j}) \in f_{i_1} \times \cdots \times f_{i_j} \mid T_M\left(\underset{p=1}{\overset{j}{[+]}}(J_{i_p})\right) = z\} = |f_{i_1} \times \cdots \times f_{i_j}|/M$ for all $z \in Z_M$.

<proof> Let $|f_{i_1} \times \cdots \times f_{i_j}| = A_{i_j}*M$, $A_{i_j} \in$ N, $A_{i_j} \ge 1$. Let $h = T_M\left[\underset{l \in q(f)}{[+]}(J_l)\right]$, where J_l denotes the specified value of field l. The remainder of the proof is similar to that of Theorem 2. □

Theorem 3 says that even though the sizes of all the unspecified fields are less than the given number of devices M, we can still guarantee optimal distribution, if (1) there exists a subset of the unspecified fields whose size of cartesian product is greater than or equal to M and (2) the records projected on these sets of fields are distributed uniformly among the M devices.

In the next section we will introduce field transformation techniques. By these field transformation techniques, Theorem 3 will be also utilized. The following paragraph exemplifies these techniques. Let $f_1 = \{0, 1\}$, $f_2 = \{0, 1, 2, 3, 4, 5, 6, 7\}$ and M = 16. As we discussed, Basic FX distribution method does not give an optimal distribution for this file system. This is because all the field values of f_1 and f_2 are smaller than M-1. Let X be some mapping such that $X(f_1) = \{0, 8\}$. When we apply Basic FX distribution method for $X(f_1) \times f_2$, the distribution becomes perfect optimal. (It can be easily verified by substituting $(1000)_B$ for $(001)_B$ in f_1 column of Table 1.) Now, the problem is to find a general one-to-one mapping, X, such that Basic FX distribution method for $X(f_1) \times f_2$ gives optimal distribution. It will be shown that for any values of $|f_1|$, $|f_2|$ and M, such mapping can be easily found.

We will present several field transformation techniques which produce the mapping, X, described above. Even though the techniques developed in this paper may not achieve perfect optimal distribution in all the cases, this extended FX distribution method will give strict optimal distribution for a large class of partial match queries.

4. FX Distribution With Field Transformation Functions

In the previous section Basic FX distribution was defined. In this section we extend the FX distribution method by using the field transformation techniques.

Let $f_1 \times f_2 \times \ldots \times f_n$ be a set of all buckets. Extended FX distribution method allocates bucket $<J_1, \ldots, J_n>$, $J_j \in f_j$ for j = 1, ... , n, into device $T_M\left[\underset{j=1}{\overset{n}{[+]}}(X^{M,|f_j|}(J_j))\right]$, where

i) if $|f_j| \ge$ M, $X^{M,|f_j|}$ is an identity function,

ii) if $|f_j| <$ M, $X^{M,|f_j|}$ is an element of set of injective (one-to-one) functions whose domains are f_j and ranges are Z_M.

$X^{M,|f_j|}$ is called *field transformation function.*

When $X^{M,|f_j|}$ is identity function for all j=1, ... , n, Extended FX distribution method reduces to Basic FX distribution method. In the following subsections we describe field transformation functions which are used for fields whose sizes are less than the given number of devices. From now on, we will simply call FX distribution instead of Extended FX distribution.

It is easy to see that all lemmas and theorems that hold for Basic FX distribution also hold for FX distribution. Since the fields whose sizes are no less than the given number of devices M, do not cause any problem (whether it is specified or not), from now on we will focus only on fields whose sizes are less than M.

4.1. Field Transformation Functions

Definition : Let M be a power of 2.

(1) I : N → N is an identity function.

(2) For a proper subset f_l of Z_M, where $|f_l|$ is some power of 2, $U^{M,|f_l|} : f_l \to Z_M$ is a function such that $U^{M,|f_l|}(l) = l d_l^{M,|f_l|}$, where $l \in f_l$, $d_l^{M,|f_l|} = \dfrac{M}{|f_l|}$.

(3) For a proper subset f_l of Z_M, where $|f_l|$ is some power of 2, $\mathbf{IU1}^{M,|f_l|} : f_l \rightarrow Z_M$ is a function such that $\mathbf{IU1}^{M,|f_l|}(l) = l$ [+] $ld_l^{M,|f_l|}$, where $l \in f_l$, $d_l^{M,|f_l|} = \frac{M}{|f_l|}$.

(4) For a proper subset f_l of Z_M, where $|f_l|$ is some power of 2, $\mathbf{IU2}^{M,|f_l|} : f_l \rightarrow Z_M$ is a function such that $\mathbf{IU2}^{M,|f_l|}(l) = l$ [+] $ld_{l1}^{M,|f_l|}$ [+] $ld_{l2}^{M,|f_l|}$, where $l \in f_l$, $d_{l1}^{M,|f_l|} = \frac{M}{|f_l|}$, $d_{l2}^{M,|f_l|} = \begin{cases} d_{l1}^{M,|f_l|}/|f_l| & \text{if } |f_l|^2 < M \\ 0 & \text{otherwise} \end{cases}$

We have defined the four groups of basic functions, I, $U^{M,|f_l|}$, $IU1^{M,|f_l|}$ and $IU2^{M,|f_l|}$ which will be used in various combinations for optimal file distribution. For example, for any values of $|f_i|$, $|f_j|$ and M, it will be shown that FX distribution method distributes elements of $I(f_i) \times U^{M,|f_j|}(f_j)$ optimally.

It is easy to see that for any proper subset f_l of Z_M whose $|f_l|$ is some power of 2, I and $U^{M,|f_l|}$ satisfies the requirements of field transformation functions described previously. We will show later that $IU1^{M,|f_l|}$ and $IU2^{M,|f_l|}$ satisfy these requirements. The operation of $X^{M,|f_l|}$ function on field l is said to be X transformation of field l. For example, when $U^{M,|f_l|}$ function is applied to field l, we say that U transformation is used for field l. Two transformation functions, $X1^{M1,J}$ and $X2^{M2,J}$ are said to be the *same transformation method*, if X1 = X2. For example, $U^{M1,|f_i|}$ and $U^{M2,|f_j|}$ are the same transformation method. But $U^{M1,|f_i|}$ and $IU1^{M2,|f_j|}$ are said to be *different transformation methods*.

Because of notational complexity, when the context is clear, we will leave out the superscripts M, $|f_l|$ from transformation functions and their parameters.

In section 4.1.1 properties of I, U and IU1 transformation are explained and their related strict optimality is described. In section 4.1.2 IU2 transformation and its related strict optimality with I and U transformation is described.

4.1.1. FX Distribution With I, U and IU1 Transformation

From the definition of U transformation we see that the transformed domain elements are equally spaced. In other words, when they are linearly ordered, any two adjacent elements will have the same distance. The property of IU1 transformation will be characterized by Lemma 5.4. In the following subsections, relationships between I, U and IU1 transformation will be investigated.

4.1.1.1. Strict Optimal By I and U Transformation

Lemma 4.1. Let L = aw + b, a≥0, 0 ≤ b ≤ w-1, where w is some power of 2. Let W = {0, 1, ... , w-1}. Then, W [+] L = {aw, aw+1, ... , (a+1)w-1}.

<proof> Given in [KiPr87].

Theorem 4. When there are only two fields i, j whose sizes are less than the given number of devices M, FX distribution with $I(f_i)$ and $U(f_j)$ is perfect optimal.

<proof> Let $d_j = M/F_j$.

(case 1) $F_iF_j < M$ or $d_j > F_i$

$I(f_i)$ [+] $U(f_j)$ = {0, 1, ... , F_i-1} ∪ {d_j, d_j+1, ... , d_j+F_i-1} ∪ ... {M-d_j, M-d_j+1, ... , M-d_j+F_i-1} by Lemma 4.1. All sets in the right hand side are disjoint and the largest element in these sets is less than M because $d_j > F_i$. So, for all $z \in Z_M$, #{$(J_i,J_j) \in f_i \times f_j \mid I(J_i)$ [+] $U(J_j) = z$} ≤ 1. Therefore, it is 2-optimal. 0 and 1-optimal come from Theorem 1.

(case 2) $F_iF_j \geq M$ or $d_j \leq F_i$

Let $F_iF_j = A*M$ or $d_j = F_i/A$. In order to be 2-optimal, for all $z \in Z_M$, #{$(J_i,J_j) \in f_i \times f_j \mid I(J_i)$ [+] $U(J_j) = z$} should be equal to A.

1) $0 \leq U(J_j) \leq (A-1)d_j$

For each $U(J_j)$ within this range, $I(f_i)$ [+] $U(J_j)$ = {0, 1, ... , F_i-1} by Lemma 4.1. Let this set be S_0. Since there are A number of such $U(J_j)$ within this range, for all $s \in S_0$ #{$(J_i,J_j) \in f_i \times f_j \mid 0 \leq U(J_j) \leq (A-1)d_j$, $I(J_i)$ [+] $U(J_j) = s$} = A.

2) $Ad_j \leq U(J_j) \leq (2A-1)d_j$

For each $U(J_j)$ within this range, $I(f_i)$ [+] $U(J_j)$ = {F_i, F_i+1, ⋯ , 2F_i−1} by Lemma 4.1. Let this set be S_1. Since there are A number of such $U(J_j)$ within this range, for all $s \in S_1$ #{$(J_i,J_j) \in f_i \times f_j \mid Ad_j \leq U(J_j) \leq (2A-1)d_j$, $I(J_i)$ [+] $U(J_j) = s$} = A.

. . .

. . .

$\frac{M}{F_i}$) $(\frac{M}{F_i}-1)Ad_j \leq U(J_j) \leq M-d_j$

For each $U(J_j)$ within this range, $I(f_i)$ [+] $U(J_j)$ = {$(\frac{M}{F_i}-1)F_i$, $(\frac{M}{F_i}-1)F_i$+1, ... M−1}. Let this set be $S_{\frac{M}{F_i}-1}$. Since there are A number of such $U(J_j)$ within this range, for all $s \in S_{\frac{M}{F_i}-1}$, #{$(J_i,J_j) \in f_i \times f_j \mid (\frac{M}{F_i}-1)Ad_j \leq U(J_j) \leq M-d_j$, $I(J_i)$ [+] $U(J_j) = s$} = A.

Since $\bigcup_{p=0}^{\frac{M}{F_i}-1} S_p = Z_M$ and there are A repetitions for each element in Z_M through $I(f_i)$ [+] $U(J_j)$, J_j = 0, ⋯ F_j−1, it is 2-optimal. 0 and 1-optimal come from Theorem 1. □

Example 3. Let f_1 = {0, 1, 2, 3}, f_2 = {0, 1, 2, 3} and M = 16. Table 2 shows bucket distribution by FX and Modulo methods. Note that $I(f_1)$ = {0, 1, 2, 3} and $U(f_2)$ = {0, 4, 8, 12} are I and U transformed values of f_1, f_2 and denoted by binary numbers. Here, Device No = $T_M(I(J_1)$ [+] $U(J_2))$ for FX distribution, and Device No = $(J_1 + J_2)$ *mod M* for Modulo distribution, where $J_1 \in f_1, J_2 \in f_2$.

The FX distribution in Table 2 is optimal. But in Modulo distribution, it is skewed. GDM method can also give optimal distribution by

f_1	f_2	$I(f_1)$	$U(f_2)$	Device No (FX)	Device No (Modulo)
0	0	0000	0000	0	0
0	1	0000	0100	4	1
0	2	0000	1000	8	2
0	3	0000	1100	12	3
1	0	0001	0000	1	1
1	1	0001	0100	5	2
1	2	0001	1000	9	3
1	3	0001	1100	13	4
2	0	0010	0000	2	2
2	1	0010	0100	6	3
2	2	0010	1000	10	4
2	3	0010	1100	14	5
3	0	0011	0000	3	3
3	1	0011	0100	7	4
3	2	0011	1000	11	5
3	3	0011	1100	15	6

Table 2. FX distribution with I and U transformation

multiplying 3 to the first field values and by 4 to the second field values. However, these parameters should be found by trial and error. On the other hand, FX distribution techniques give a specific method.

4.1.1.2. Strict Optimal By I and IU1 Transformation

Lemma 5.1. When F_k is less than M, $IU1(f_k)$ is a set of F_k elements between 0 and M-1. (i.e., $IU1^{M,|f_k|}(f_k)$ is an injective function)

<proof> Given in [KiPr87].

Example 4. Let f_k = {0, 1, 2, 3, 4, 5, 6, 7} and M = 16. Then $IU\,1(f_k)$ = {0, 3, 6, 5, 12, 15, 10, 9}.

Lemma 5.2. When there are only 2 fields i, k whose sizes are less than the given number of devices M and $F_i \geq F_k$, FX distribution with $I(f_i)$ and $IU\,1(f_k)$ is perfect optimal.
<proof> The proof follows from Lemma 1.1 and Lemma 4.1. The complete proof is given in [KiPr87].

Definition When F_j < M and d = $M/F_j \geq 2$, there are F_j intervals [0, d), [d, 2d), ... , [M-d, M) from 0 to M with interval size $d = \frac{M}{F_j}$. Throughout this paper, boundaries of the interval are always assumed to half-closed as above.

Lemma 5.3. When there are only two fields i and k whose $F_iF_k = M$, FX distribution with $I(f_i)$ and $IU\,1(f_k)$ is perfect optimal if and only if there is exactly one element of $IU\,1(f_k)$ at each interval of size $\frac{M}{F_k}$.
<proof> Given in [KiPr87].

Lemma 5.4. For any F_k which is less than M, there is exactly one element of $IU\,1(f_k)$ at each interval from 0 to M with interval size $d_k = \frac{M}{F_k}$.
<proof> By Lemma 5.2 and Lemma 5.3 we can conclude Lemma 5.4. The complete proof is given in [KiPr87]. □

Theorem 5. When there are only two fields i, k whose F_i and F_k are less than the given number of devices M, FX distribution with $I(f_i)$ and $IU\,1(f_k)$ is perfect optimal.
<proof> Let $d_k = \frac{M}{F_k}$. By Lemma 5.4 there is exactly one element of $IU\,1(f_k)$ at each interval from 0 to M with interval size d_k. Let $IU\,1(f_k)$ = $\{t_0, t_1, \ldots, t_{F_k-1}\}$, where $t_0 < t_1 < \cdots < t_{F_k-1}$. Then, each t_l is in the interval $[ld_k, (l+1)d_k)$. The remainder of the proof is similar to that of Theorem 4. □

Example 5. Let f_1 = {0, 1, 2, 3}, f_2 = {0, 1, 2, 3} and M = 16. Table 3 shows the bucket distribution by FX distribution with $I(f_1)$ and $IU\,1(f_2)$. Here, $IU\,1(f_2)$ = {0, 5, 10, 15} and Device No = $T_M(I(J_1)$ [+] $IU\,1(J_2))$, $J_1 \in f_1, J_2 \in f_2$.

$I(f_1)$	$IU\,1(f_2)$	Device No
0000	0000	0
0000	0101	5
0000	1010	10
0000	1111	15
0001	0000	1
0001	0101	4
0001	1010	11
0001	1111	14
0010	0000	2
0010	0101	7
0010	1010	8
0010	1111	13
0011	0000	3
0011	0101	6
0011	1010	9
0011	1111	12

Table 3. FX distribution with I and IU1 transformation

4.1.1.3. Strict Optimal By U and IU1 Transformation

Definition Let $|f_j|$ be less than the given number of devices M. When $U(f_j)$ = {0, d_j, ... , $(F_j-1)d_j$} with $d_j = \frac{M}{F_j}$, $U(f_j) + c$ is defined as {0+c, d_j+c, ... , $(F_j-1)d_j + c$}, where c < d_j, c ∈ N.

Lemma 6.1 Let f_j = {0, 1, ... , F_j-1} such that $F_j < M$. Let $d_j = \frac{M}{F_j}$. Then, for any nonnegative integer J and c such that $0 \leq J < F_j$ and 0 ≤ c < d_j, $U(f_j)$ [+] $(Jd_j + c) = U(f_j) + c$.
<proof> Given in [KiPr87].

Lemma 6.2 K_1 mod d_j = K_2 mod d_j if and only if (K_1 [+] K_1d_k) mod d_j = (K_2 [+] K_2d_k) mod d_j, where K_1, K_2 are any nonnegative integers, and d_j, d_k are any power of 2 which are greater than 1.
<proof> Given in [KiPr87].

Theorem 6. When there are only two fields j, k whose F_j, F_k are less than the given number of devices M, FX distribution with $U(f_j)$ and $IU\,1(f_k)$ is perfect optimal.
<proof> Let $d_j = \frac{M}{F_j}$ and $d_k = \frac{M}{F_k}$. Let K ∈ f_k.
(case 1) F_jF_k > M or $F_k > d_j$
Let $F_jF_k = AM$. Then $A = \frac{F_k}{d_j} = \frac{F_j}{d_k}$ or $F_k = Ad_j$.
1) K mod d_j = 0 or K = Jd for some J ∈ f_j
Since $F_k = Ad_j$, there are A number of $IU\,1(K)$ such that K mod d_j = 0. For such $IU\,1(K)$, $U(f_j)$ [+] $IU\,1(K) = U(f_j)$ [+] K [+] $Kd_k = U(f_j)$ [+] Kd_k = $U(f_j)$ by Lemma 6.1. Let this set be S_0. So, there are A repetitions for each element in S_0 through $U(f_j)$ [+] $IU\,1(K)$, $K = 0, d_j, \cdots, (A-1)d_j$.
2) K mod d_j = 1
By Lemma 6.2 all IU1(K) such that K mod d_j = 1 has the same residue c_1 by modulus d_j. So, all such IU1(K) can be represented as $\alpha d_j + c_1$ for some variable $\alpha \in Z_{F_j}$ and some fixed nonnegative integer c_1 which is less than d_j. By Lemma 6.1 $U(f_j)$ [+] $(\alpha d_j + c_1) = U(f_j) + c_1$. Let this set be S_1. Since there are A number of $IU\,1(K)$ such that K mod d_j = 1, there are A repetitions for each element in S_1 through $U(f_j)$ [+] $IU\,1(K)$, $K = 1, d_j+1, \cdots, (A-1)d_j+1$.

. . .

d_j-1) K mod $d_j = d_j - 1$
By Lemma 6.1 and 6.2, for any IU1(K) such that K mod $d_j = d_j-1$, $U(f_j)$ [+] $IU\,1(K) = U(f_j) + c_{d_j-1}$ for some fixed c_{d_j-1} such that $c_{d_j-1} < d_j$. Let this set be S_{d_j-1}. Then, there are A repetitions for each element in S_{d_j-1} through $U(f_j)$ [+] $IU\,1(K)$, $K = d_j-1, \cdots, Ad_j-1$. By Lemma 6.2, all c_is are different each other. So, $\bigcup_{l=0}^{d_j-1} S_l = Z_M$. Since there are A repetitions for each element of Z_M through $U(f_j)$ [+] $IU\,1(K)$, $K = 0, \cdots, F_k-1$, it is 2-optimal. 0 and 1-optimal come from Theorem 1.
(case 2) $F_jF_k < M$ or $F_k < d_j$
The proof is similar to that of case 1. □

4.1.1.4. Strict Optimal By I, U and IU1 Transformation

Corollary 6.1 Let $q_s(f)$ be a set of fields which are unspecified for a partial match query q and whose sizes are less than the given number of devices M. FX distribution in which I, U or IU1 transformation methods are used, is strict optimal for partial match query q, if (1) the conditions of Theorem 1 or 2 are satisfied or (2) $|q_s(f)| = 2$ and for i, j ∈ $q_s(f)$, their transformation methods are different (refer to the beginning of section 4.1) or (3) $|q_s(f)| \geq 3$ and there exist at least 2 fields i,j ∈ $q_s(f)$ such that $F_iF_j \geq M$ and their transformation methods are different from each other.
<proof> This is a direct consequence of Theorem 1, 2, 3, 4, 5 and 6. □

Example 6. Let f_1 = {0, 1}, f_2 = {0, 1, 2, 3} and f_3 = {0, 1} and M = 8. Table 4 shows the bucket distribution by FX distribution with $I(f_1)$, $U(f_2)$, $IU\,1(f_3)$. Here, $U(f_2)$ = {0, 2, 4, 6} and $IU\,1(f_3)$ = {0, 5}.

$I(f_1)$	$U(f_2)$	$IU1(f_3)$	Device No
000	000	000	0
000	000	101	5
000	010	000	2
000	010	101	7
000	100	000	4
000	100	101	1
000	110	000	6
000	110	101	3
001	000	000	1
001	000	101	4
001	010	000	3
001	010	101	6
001	100	000	5
001	100	101	0
001	110	000	7
001	110	101	2

Table 4. FX distribution with I, U and IU1 transformation

4.1.2. FX Distribution With I, U and IU2 Transformation

In this section, properties of IU2 transformation are described. The relationships between I, U and IU2 transformation are investigated in the following subsections.

Lemma 7.1. When F_k of field k is less than the given number of devices M, $IU2(f_k)$ is a set of F_k elements between 0 and M-1. (i.e., $IU2^{M,|f_k|}$ is an injective function.)
<proof> Given in [KiPr87] □

Note that when $F_k^2 \geq$ M, IU2 transformation becomes the same as IU1 transformation.

Lemma 7.2. For any F_k which is less than the given number of devices M, there is exactly one element of $IU2(f_k)$ at each interval from 0 to M with interval size $d_{k1} = \frac{M}{F_k}$. (refer to section 4.1.1.2. for the interval boundaries)
<proof> Given in [KiPr87] □

4.1.2.1. Strict Optimal By I and IU2 Transformation

Theorem 7. When there are only two fields i and k whose F_i and F_k are less than the given number of devices M, FX distribution with $I(f_i)$ and $IU2(f_k)$ is perfect optimal.
<proof> The proof is almost the same as that of Theorem 5 except that Lemma 7.2 is used instead of Lemma 5.4. □

Example 7. Let $f_1 = \{0, 1, 2, 3, 4, 5, 6, 7\}$, $f_2 = \{0, 1\}$ and M = 16. Table 5 shows the FX distribution with $I(f_1)$ and $IU2(f_2)$. Here, $IU2(f_2) = \{0, 13\}$.

$I(f_1)$	$IU2(f_2)$	Device No
0000	0000	0
0000	1101	13
0001	0000	1
0001	1101	12
0010	0000	2
0010	1101	15
0011	0000	3
0011	1101	14
0100	0000	4
0100	1101	9
0101	0000	5
0101	1101	8
0110	0000	6
0110	1101	11
0111	0000	7
0111	1101	10

Table 5. FX distribution with I and IU2 transformation

4.1.2.2. Strict Optimal By U and IU2 Transformation

Lemma 8.1. Let $f_k = \{0, 1, \ldots, F_k-1\}$ and $F_k < M$. Then, for $K_1, K_2 \in f_k$, $K_1 \bmod d_j = K_2 \bmod d_j$ if and only if $(K_1 [+] K_1 d_{k2} [+] K_1 d_{k1}) \bmod d_j = (K_2 [+] K_2 d_{k2} [+] K_2 d_{k1}) \bmod d_j$, where d_{k1} and d_{k2} are parameters of IU2 transformation (i.e., $d_{k1} = \frac{M}{F_k}$. $d_{k2} = d_{k1}/F_k$ if $F_k^2 <$ M, and 0 otherwise), and d_j is any power of 2 which is greater than 1.
<proof> Given in [KiPr87].

Theorem 8. When there are only two fields j and k whose F_j and F_k are less than the given number of devices M, FX distribution with $U(f_j)$ and $IU2(f_k)$ is perfect optimal.
<proof> The proof is almost the same as that of Theorem 6, except that Lemma 8.1 is used instead of Lemma 6.2. □

4.1.2.3. Strict Optimal By I, U and IU2 Transformation

Lemma 9.1. When there are only three fields i, j and k whose sizes are less than the given number of devices M, FX distribution with $I(f_i)$, $U(f_j)$ and $IU2(f_k)$ is perfect optimal, if either
(1) there exist at least 2 fields p and q such that p,q $\in$ {i, j, k} and $F_pF_q \geq M$ or
(2) $F_k \geq F_j$ and $F_k^2 < M$
<proof> It is clear that the first condition given in Lemma 9.1 satisfies perfect optimality condition by Theorem 1, 3, 4, 7 and 8. So, let us consider only the other case, that is, $F_pF_q < M$ for any p and q, p,q $\in$ {i, j, k}. We want to show that second condition given in Lemma 9.1 is sufficient for perfect optimal distribution. Let $d_j = M/F_j$ and $d_{k1} = M/F_k$ and $d_{k2} = d_{k1}/F_k$. Then $d_j > F_i$, $d_j > F_k$ and $d_j > d_{k1}$ and $d_{k1} > F_i$, $d_{k1} > F_j$. Since $F_k^2 < M$, $d_{k1} > F_k$ or $d_{k2} > 1$.
(case 1) $F_iF_jF_k \geq M$
Let $F_iF_jF_k = AM$, and let $d_j = Bd_{k1}$ and $d_{k1} = CF_i$, where A,B,C $\in$ N and A,B,C ≥ 1. Then $F_i\frac{M}{d_j}F_k = AM$ or $F_i = A\frac{d_j}{F_k} = ABd_{k2}$. Since $d_j > F_i$, $I(f_i)$ [+] $U(f_j) = S_0 \cup S_1 \cup \ldots \cup S_{F_j-1}$, where

$$S_0 = \{0, 1, \ldots, F_i-1\}$$
$$\cdots$$
$$S_i = \{id_j, id_j+1, \ldots, (i+1)d_j+F_i-1\}$$
$$\cdots$$
$$S_{F_j-1} = \{(F_j-1)d_j, (F_j-1)d_j+1, \ldots, (F_j-1)d_j+F_i-1\}$$

It is clear that all S_is are disjoint. For such S_i as above, $S_i+c = \{id_j+c, id_j+1+c, \ldots, id_j+F_i-1+c\}$, where $0 \leq c < d_j$ and c $\in$ N. (This is defined in section 4.1.1.3) Let K $\in f_k$. By Lemma 5.4 there is exactly one element of K [+] Kd_{k2} at each interval from 0 to d_{k1} with interval size d_{k2}.
1) $0 \leq K$ [+] $K\ d_{k2} < F_i$
Since F_i = ABd_{k2}, there are AB number of $IU2(K)$ such that $0 \leq K$ [+] $K\ d_{k2} < F_i$.
1-1) K mod B = 0 or $Kd_{k1} \bmod d_j = 0$
For each $IU2(K)$ within this range, $I(f_i)$ [+] $U(f_j)$ [+] $IU2(K) = \bigcup_{i=0}^{F_j-1} S_i$
The equality holds due to Lemma 1.1. (Note that K [+] Kd_{k1} [+] Kd_{k2} = Kd_{k1} [+] (K [+] Kd_{k2}) Let this set be T_{11}. Since there are A number of such $IU2(K)$s within this range, there are A repetitions for each element in T_{11} through $I(f_i)$ [+] $U(f_j)$ [+] $IU2(K)$, for all $IU2(K)$ within this range.
1-2) K mod B = 1 or $Kd_{k1} \bmod d_j = d_{k1}$
For each $IU2(K)$ within this range, $I(f_i)$ [+] $U(f_j)$ [+] $IU2(K) = \bigcup_{i=0}^{F_j-1} S_i + d_{k1}$. The equality holds due to Lemma 1.1 and Lemma 4.1. (Note that $d_{k1} < d_j$.) Let this set be T_{12}. Since there are A number of such $IU2(K)$ elements within this range, there are A repetitions for each element in T_{12}

. . .

1-B) K mod B = B − 1 or $Kd_{k1} \bmod d_j = (B-1)d_{k1}$

For each $IU2(K)$ within this range, $I(f_i)$ [+] $U(f_j)$ [+] $IU2(K) = \bigcup_{i=0}^{F_j-1} S_i + (B-1)d_{k1}$. Let this set be T_{1B}. There are A repetitions for each element in T_{1B}.

2) $F_i \le K$ [+] $Kd_{k2} < 2F_i$

2-1) K mod B = 0

For each $IU2(K)$ within this range, $I(f_i)$ [+] $U(f_j)$ [+] $IU2(K) = \bigcup_{i=0}^{F_j-1} S_i + F_i$. Let this set be T_{21}. There are A repetitions for each element in T_{21}.

. . .

2-B) K mod B = B − 1

For each $IU2(K)$ within this range, $I(f_i)$ [+] $U(f_j)$ [+] $IU2(K) = \bigcup_{i=0}^{F_j-1} S_i + (B-1)d_{k1} + F_i$. (Note that $d_{k1} > F_i$.) Let this set be T_{2B}. There are A repetitions for each element in T_{2B}.

. . .

C) $(C-1)F_i \le K$ [+] $Kd_{k2} < CF_i$

C-1) K mod B = 0

For each $IU2(K)$ within this range, $I(f_i)$ [+] $U(f_j)$ [+] $IU2(K) = \bigcup_{i=0}^{F_j-1} S_i + (C-1)F_i$. (Note that $CF_i = d_{k1}$.) Let this set be T_{C1}. There are A repetitions for each element in T_{C1}.

. . .

C-B) K mod B = B - 1

For each $IU2(K)$ within this range, $I(f_i)$ [+] $U(f_j)$ [+] $IU2(K) = \bigcup_{i=0}^{F_j-1} S_i + (B-1)d_{k1} + (C-1)F_i$. Let this set be T_{CB}. There are A repetitions for each element in T_{CB}. Now, it is not difficult to see that $\bigcup_{i=1,j=1}^{i=C,j=B} T_{ij} = Z_M$ and any two different T_{ij}s are disjoint. Since we already show that there are A repetitions for every element in Z_M, it is 3-optimal. 2-optimal come from Theorem 4, 7, 8. 0 and 1-optimal come from Theorem 1.

(case 2) $F_iF_jF_k < M$

The proof is similar to that of case 1. □

Theorem 9. Let L be the set of fields whose sizes are less than the number of devices M in a given file system. FX distribution with I, U and IU2 transformation can be always perfect optimal, if $|L| \le 3$.

<proof> The result follows from Theorem 1, 2, 3, 4, 7 and 8 when L = 0, 1, 2. When L = 3, let i, j, k be fields whose sizes are less than M and $F_i \ge F_k \ge F_j$. Apply $I(f_i)$, $U(f_j)$ and $IU2(f_k)$ transformation. If $F_k^2 \ge M$, then $F_iF_k \ge M$. So, Theorem 9 holds by the first condition in Lemma 9.1. If $F_k^2 < M$, Theorem 9 holds by the second condition of Lemma 9.1. □

Example 8. Let $f_1 = \{0, 1, 2, 3\}$, $f_2 = \{0, 1\}$, $f_3 = \{0, 1\}$ and M = 16. Table 6 shows the FX distribution with $I(f_1)$, $U(f_2)$ and $IU2(f_3)$.

$I(f_1)$	$U(f_2)$	$IU2(f_3)$	Device No
0000	0000	0000	0
0000	0000	1101	13
0000	1000	0000	8
0000	1000	1101	5
0001	0000	0000	1
0001	0000	1101	12
0001	1000	0000	9
0001	1000	1101	4
0010	0000	0000	2
0010	0000	1101	15
0010	1000	0000	10
0010	1000	1101	7
0011	0000	0000	3
0011	0000	1101	14
0011	1000	0000	11
0011	1000	1101	6

Table 6. FX distribution with I, U and IU2 transformation

Corollary 9.1 Let $q_s(f)$ be the set of fields which are unspecified for some partial match query q and whose sizes are less than M. FX distribution in which I, U or IU2 transformation methods are used, is strict optimal for partial match query q, if (1) the conditions of Theorem 1 or 2 are satisfied or (2) $|q_s(f)| = 2$ and for i, j $\in q_s(f)$, their transformation methods are different (refer to the beginning of section 4.1) or (3) there exist 2 fields i,j $\in q_s(f)$ such that $F_iF_j \ge M$ and their transformation methods are different or (4) $|q_s(f)| = 3$ and transformation methods of these three fields are different and the second conditions of Lemma 9.1 for these fields are satisfied or (5) when $|q_s(f)| \ge 4$, there exist 3 fields i, j, k $\in q_s(f)$ such that $F_iF_jF_k \ge M$ and the second conditions of Lemma 9.1 for these fields are satisfied.

<proof> This is a direct consequence of Theorem 1, 2, 3, 4, 7, 8 and 9. □

4.2. Summary And Discussions

In section 3 and section 4.1 we determined, through theorems and corollaries, the class of partial match queries whose qualified buckets are distributed optimally under FX distribution. The results of FX distribution are summarized as following :

Let a file consisting of n fields be distributed among M parallel devices. Let L be the number of fields whose sizes are less than the given number of devices M. FX distribution can be always perfect optimal, when $L \le 3$. Let L be greater than or equal to 4. Let $q(f)$ be the set of fields which are unspecified for partial match query q. Then, FX distribution is strict optimal for partial match query q, if at least one of the following conditions holds.

(1) $|q(f)| = 0$ or 1

(2) there is at least one unspecified field $p \in q(f)$ such that $F_p \ge M$.

(3) $|q(f)| = 2$ and transformation methods of two fields in $q(f)$ are different.

(4) $|q(f)| = 3$ and either

  (a) there are at least two fields p, q $\in q(f)$ such that $F_pF_q \ge M$ and transformation methods of two fields p and q are different, or

  (b) transformation methods of three fields in $q(f)$ are I, U, IU2 and the size of IU2 transformed field is not less than the size of U transformed field.

(5) $|q(f)| \ge 4$ and either

  (a) there are at least two fields p, q $\in q(f)$ such that $F_pF_q \ge M$ and transformation methods of two fields p and q are different, or

  (b) there are at least three fields i, j, k $\in q(f)$ such that $F_iF_jF_k \ge M$ and transformation methods of fields i, j and k are I, U, IU2 and the size of IU2 transformed field is not less than the size of U transformed field.

Here, IU2 transformation does not apply for the field whose square of the field size is greater than or equal to M, and in (3), (4)-a and (5)-a IU1 and IU2 combination do not apply.

When the number of devices and the size of each field are power of 2, the set of partial match queries which are optimal under FX distribution is a superset of those for Modulo distribution [DuSo82].

It is unfortunate that FX Distribution does not always guarantee strict optimal distribution when the number of fields whose sizes are less than M, is greater than or equal to 4 in general. In fact, it has been shown in [Sung87] that when the number of fields whose sizes are less than the given number of devices, is greater than or equal to 4, there is no method which always gives perfect optimal distribution. However, even for such

cases FX distribution gives strict optimal distribution for a large class of partial match queries.

For main memory databases implemented on multiprocessor systems, bucket distribution and *inverse mapping* should be fast, where inverse mapping is a procedure used for each device to find qualified buckets residing in it. This is because each device has only a subset of total qualified buckets. If the distribution method is complex, not only it takes time to distribute buckets, but it also takes more time for inverse mapping. So, complexity of distribution method should be an important criterion for main memory database systems. We claim that FX distribution gives fast response time in main memory databases because the computations for bucket distribution and inverse mapping is very simple.

5. Performance Comparisons to Other Distribution Methods

In this section we compare FX distribution with Modulo and GDM method. The performance comparisons are based on the probability of strict optimality and response time for a given partial match query. In section 5.1 the probability of strict optimal distribution for partial match queries is compared. In section 5.2 we compare the average response time for FX, Modulo and GDM distribution.

For both section 5.1 and 5.2, it is assumed that the probability of each field being specified is same for all fields and some field being specified is independent of each other.

5.1. Probability of Strict Optimality

In this section we show that the probability of strict optimality for FX distribution is much better than Modulo distribution. Even for the worst case the decrease of probability of strict optimality for FX distribution is not much. On the other hand, in Modulo distribution the decrease is quite large. Since no general method has been given to determine the existence of parameter values for strict optimal distribution in GDM method, we compare FX distribution to only Modulo distribution in this section.

Figure 1 through 4 shows the percentage of strict optimal distribution for all possible partial match queries in a given file system. In all these Figures MD denotes Modulo Distribution and FD denotes FX Distribution. Here, results are computed from sufficient conditions given for each method. Figure 1 and 2 show the case where any two fields p and q satisfy the condition, $F_pF_q \geq M$. The number of fields for Figure 1 and 2 are 6 and 10, respectively. In Figure 1 and 2 FX distribution used I, U and IU1 transformation methods.

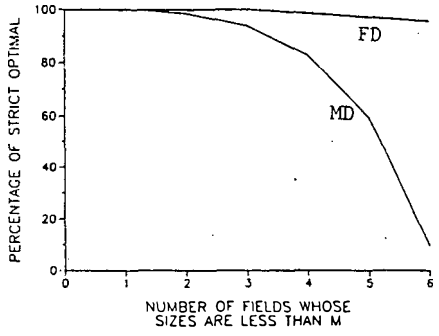

Figure 1

Figure 3 and 4 show the percentage of strict optimal distribution when for any two fields p,q, $F_pF_q < M$ but for any three fields p,q,r $F_pF_qF_r \geq M$. The number of fields for Figure 3 and 4 are 6 and 10, respectively. Here, in FX distribution I, U and IU2 transformation methods are used.

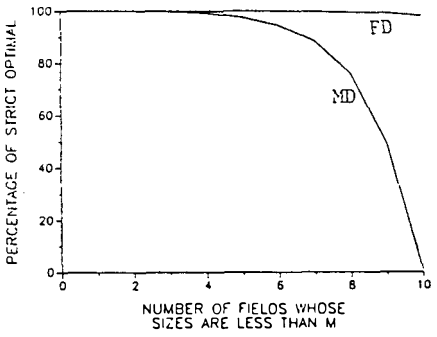

Figure 2

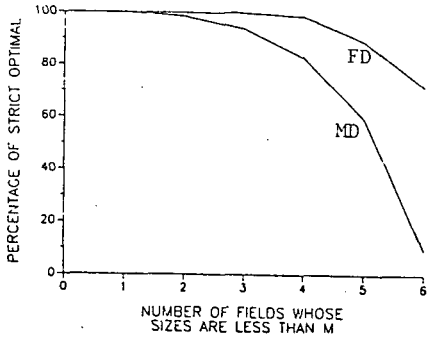

Figure 3

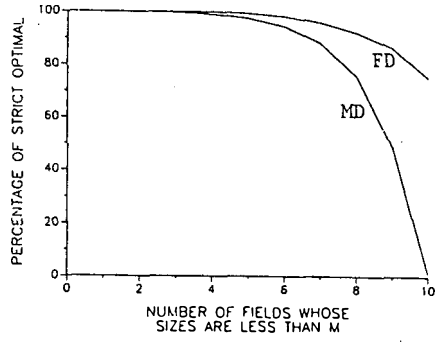

Figure 4

5.2. Average Response Time

Definition : For a given partial match query q, $r_i(q)$ is defined as the number of qualified buckets in device i for a partial match query q. We call this a *response size* for device *i*. Then, a *largest response size* for a partial match query q is defined as $MAX(r_1(q), r_2(q), \cdots r_{M-1}(q))$.

We will consider two factors for the response time of a partial match query. They are the largest response size and CPU computation time for bucket distribution and inverse mapping. In parallel disks environment, largest response size is the most important factor, while in main memory databases, CPU computation time is more important.

In section 5.2.1, comparisons based on the largest response size are described, and in section 5.2.2, the CPU computation time is discussed.

5.2.1. Largest Response Size

In this section, it is assumed that all parallel devices have the same characteristics, and the interconnection network topology is symmetric. In other words, systems are configured such that the data retrieval time for

any device is almost the same. For example, parallel disks connected to one shared bus or multiprocessors based on multistage interconnection networks are considered to have symmetric network topology. Then, the response time for a partial match query is determined by the device which has the largest number of qualified buckets.

Table 7 through 9 show the largest response size of Modulo, GDM and FX distribution for some typical file system environments. The number of fields is assumed to be 6 for all these experiments. The first column denotes the number of unspecified fields. For GDM method, in order for comparisons to be fair, we used three different sets of multiplication parameters. These sets are GDM1 : 2, 3, 5, 7, 11, 13 and GDM2 : 2, 5, 11, 43, 51, 57 and GDM3 : 41, 43, 47, 51, 53, 57. The FX distribution of Table 7 and 8 used I transformation for fields 1 and 4, U transformation for fields 2 and 5, IU1 transformation for fields 3 and 6. The FX distribution of Table 9 used IU2 transformation instead of IU1 transformation and others are the same as in Table 7 and 8. Each entry in the tables is computed by averaging values of largest response sizes from all possible partial match queries for that entry.

	Modulo	GDM1	GDM2	GDM3	FX	Optimal
2	8.0	3.3	3.6	3.7	3.2	2.0
3	48.0	18.1	18.9	18.9	16.0	16.0
4	344.0	130.5	132.7	132.5	128.0	128.0
5	2460.0	1026.3	1029.7	1031.7	1024.0	1024.0
6	18152.0	8196.0	8198.0	8202.0	8192.0	8192.0

Table 7. M = 32, $F_1 = \ldots = F_6 = 8$

	Modulo	GDM1	GDM2	GDM3	FX	Optimal
2	8.0	2.1	2.2	2.4	2.4	1.0
3	48.0	10.2	10.3	10.6	8.0	8.0
4	344.0	68.3	68.1	67.5	64.0	64.0
5	2460.0	520.5	517.0	517.3	512.0	512.0
6	18152.0	4114.0	4102.0	4102.0	4096.0	4096.0

Table 8. M = 64, $F_1 = \ldots = F_6 = 8$

	Modulo	GDM1	GDM2	GDM3	FX	Optimal
2	9.6	1.7	1.3	1.4	2.3	1.0
3	91.2	10.0	5.5	5.6	5.1	3.2
4	911.2	90.3	40.5	42.2	37.3	35.2
5	9076.0	909.5	397.3	408.67	384.0	384.0
6	90404.0	9176.0	4144.0	4313.0	4096.0	4096.0

Table 9. M = 512, $F_1 = F_2 = F_3 = 8$ and $F_4 = F_5 = F_6 = 16$

The tables show that except for first row of table 8 and 9, FX distribution gives smaller largest-response-size than the other methods. FX distribution is also very close to optimal. It should also be noted that there may be a set of multiplication parameters by which GDM method can give better performance than those of GDM1, GDM2 and GDM3. Even though such a set of parameters may exist, it can only be found by trial and error method.

5.2.2. CPU Computation Time

In this section we compare CPU computation time for FX and GDM distribution. If environments are disk based, the computation time is usually not much significant compared to disk access time. But in main memory databases CPU computation time is quite significant.

We use optimized instruction codes for comparing CPU computation time. In GDM method we use AND operation to implement modulo function. This is possible because the number of devices is assumed to be a power of 2. In FX distribution, since the multipliers for U, IU1 and IU2 transformation are always power of 2, we can substitute multiplication by shift operation. Note that we cannot do this in GDM method because multipliers in GDM method are usually chosen from prime or odd numbers. Function T_M is done by AND operation.

In MC68000 processor, computation time of FX method takes about only one third of that of GDM method. (In MC68000, XOR takes 8 cpu clock cycles, ADD takes 4 clock cycles, AND takes 4 clock cycles, n bit shift takes 6 + 2n clock cycles. But multiplication takes 70 clock cycles). In intel 80286/80386 processor the ratios of clock cycles between different operations are almost similar to those of MC68000.

For main memory databases implemented on MIMD machines FX method is much faster than GDM method. The computation time for Modulo distribution is shorter than FX distribution. But as discussed in previous section, the Modulo distribution is not suitable when a large number of parallel devices is used.

6. Conclusion

In this paper we present file distribution method called FX distribution for partial match retrieval type queries. Here, we show several useful characteristics of exclusive-or operation for optimal data distribution. We developed four field transformation techniques for those fields whose sizes are less than the given number of parallel devices. We show that FX distribution gives strict optimal distribution for a large class of partial match queries.

Performance of FX distribution method is compared with those of the other distribution methods such as Modulo and GDM in some typical file systems. We show that FX distribution gives better probability of strict optimality than Modulo distribution. We also compare the query response time of FX method with those of others based on the number of accesses and address computation time. We show that FX distribution gives better performance than other methods.

However, current FX distribution does not guarantee strict optimal distribution when the number of parallel devices are quite large and all field sizes are much smaller than the number of parallel devices. We are developing more general transformation functions to achieve optimal data distribution for much larger class of partial match queries in more general file systems.

7. References

[1] Aho, A.V. and Ullman, J.D., "Optimal Partial-Match Retrieval When Fields Are Independently Specified," *ACM Trans. Database Systems*, vol. 4 no. 2 , June 1979, pp. 168-179.

[2] Bolour, A., "Optimality Properties of Multiple-key Hashing Functions," *JACM, vol. 26, no. 2, April 1979, pp. 196-210.*

[3] Burkhard, W.A., "Hashing and Trie Algorithms for Partial Match Retrieval," *ACM Trans. Database Systems*, vol. 4, no. 2, June 1976, pp. 175-187.

[4] Burkhard, W.A., "Partial-Match Hash Coding : Benefits of Redundancy," *ACM Trans. Database Systems*, vol.4, no.2, June 1979, pp. 228-239.

[5] Chang,C.C., Lee,R.C.T. and Du,H.C., "Some properties of Cartesian Product Files," *Proc. ACM SIGMOD Conf.* May, 1980, pp. 157-168.

[6] Crowther,W., Goodhue,J., Starr,E., Thomas,R., Millihen,W., Blackadar,T., "Performance Measurements On A 128-Node Butterfly Parallel Processor," *Proc. Int'l Conf. on Parallel Processing*, Aug. 1985, pp. 531-540.

[7] Du, H.C., "Concurrent Disk Accessing for Partial Match retrieval," *Proc. Int'l Conf. on Parallel Processing*, Aug. 1982, pp. 211-218.

[8] Du, H.C.,"On the File Design Problem for Partial Match Retrieval," *IEEE Trans. on Software Eng.*, vol. SE-11, no. 2, Feb. 1985, pp. 213-222.

[9] Du, H.C. and Sobolewski, J.S., "Disk Allocation for Cartesian Pro-

duct Files on Multiple-Disk Systems," *ACM Trans. Database Systems*, vol. 7 no. 1, March 1982, pp. 82-101.

[10] Fagin, R., "Extendible Hashing - A Fast Access Method For Dynamic Files,"*ACM Trans. Database Systems*, vol. 4 no. 3, Sept. 1979, pp. 315-344.

[11] Fang, M.T., Lee, R.C.T. and Chang,C.C., "The idea of De-clustering and Its Applications," *Proc. Conf. on Very Large Data Bases,"* Aug. 1986, pp. 181-188.

[12] Kim, M.H. and Pramanik, S.,"Optimal Data Distribution for Partial Match Retrieval," Technical Report, Computer Science Department, Michigan State University, 1987.

[13] Larson, P., "Dynamic Hashing," BIT, 1978, pp. 184-201.

[14] Larson, P., "Linear Hashing With Partial Expansions," *Proc. 6th VLDB*, 1980, pp. 224-232.

[15] Leland, M.D. and Roome, W.D., "The Silicon Database Machine" Database Machines, Fourth International Workshop, 1985, pp. 169-189.

[16] Litwin, W., "Linear Hashing : A New Tool for File and Table Addressing," *Proc. 6th VLDB*, 1980, pp. 212-223.

[17] Pramanik, S., "Performance Analysis of a Database Filter Search Hardware," *IEEE Trans. on Computers*, vol.35, no.12, Dec. 1986, pp. 1077-1082.

[18] Pramanik, S. and Davis, H., "Multi Directory Hashing," Technical Report, Computer Science Department, Michigan State University, 1986.

[19] Pramanik, S. and Kim, M.H., "HCB_tree : A B_tree Structure for Parallel Processing," *Proc. Int'l Conf. on Parallel Processing*, Aug. 1987, pp. 140-146.

[20] Pramanik, S. and Kim, M.H.,"Generalized Parallel Processing Models for Database Systems," *Proc. Int'l Conf. on Parallel Processing*, 1988.

[21] Rivest, R.L., "Partial-Match Retrieval Algorithms," *SIAM J. Computing*, vol.5, No.1, March 1976, pp. 19-50.

[22] Rosenau, T. and Jajodia, S., "Parallel Relational Database Operations on the Butterfly Parallel Processor : Projection Results," Technical Report, Naval Research Laboratory, July 1987.

[23] Rothnie, J.B.Jr. and Lozano, T.,"Attribute Based File Organization in a Paged Memory Environment," *Comm. ACM*, vol.17, no.2, 1974, pp.63-69.

[24] Stanley Y.W.Su, L.H. Nguyen, A. Eman and G. J. Lipovski, "The Architectural Features and Implementation Techniques of multicell CASSM," *IEEE Trans. on Computers*, vol. C-28, no.6, June 1979, pp. 430-445.

[25] Sung, Y.Y., "Parallel Searching for Binary Cartesian Product Files," *Proc. ACM ASC Conf.* March 1985, pp. 163-172

[26] Sung, Y.Y.,"Performance Analysis of Disk Modulo Allocation Method for Cartesian Product Files," *IEEE Trans. on Software Eng.*, vol. SE-13, no. 9, Sept. 1987, pp. 1018-1026.

[27] *Butterfly Parallel Processor Overview*, BBN Report No. 6148 version 1, March 6, 1986.