A multikey search query has many qualified records because it specifies some of the attributes and the rest of them are unspecified. Thus, data distribution among the parallel devices is important to enhance access concurrency for multikey search queries. Though there are some heuristics on data distribution for multikey accesses, range specification in a query has not been considered in those methods. In this paper we investigate optimal data distribution for multiattribute range queries in parallel processing file systems. We show for various types of multiattribute range queries that there are inherent limitations to achieve optimal distribution. The results show that optimal distribution does not exist in many cases even for files with two fields. We give sufficient conditions for the nonexistence of perfect optimal distribution for certain types of multiattribute range queries. We have also developed data distribution methods for several useful multiattribute range queries. Sufficient conditions for optimal distribution by these proposed data distribution methods are given. It will be shown that the proposed distribution methods are perfect optimal for certain types of multiattribute range queries, and strict optimal for a large class of multiattribute range queries.

Received November 1989, revised September 1991

1. INTRODUCTION

There will be many applications for large databases that cannot be performed in an acceptable response time by current database systems. Since one way of improving performance significantly is through parallelism, the importance of parallel processing in database systems has been increasingly recognized. There have been several proposals in the past for parallel processing of database systems. These include two stage processing models,8 parallel query processing strategy for complex objects6 and machine architectures for parallel processing of database systems.6,7,10

A multiattribute file is a collection of records each of which is defined as an ordered n-tuple \((r_0, \ldots, r_{n-1})\) of \(n\) values which are the keys or attributes of the records. When a query is allowed to specify conditions on more than one attributes, the search performed for this query is called multikey searching, or associative searching. Multikey searching is used in queries such as partial match retrieval and multiattribute range queries. A multiattribute range query, also called a region query or orthogonal range query, is an intersection query in which a multiplicity of the attributes are allowed to be range specified in a query’s qualification. For a multiattribute range query, a set of records satisfy the given qualification. Thus, when a file is constructed on parallel devices, it is important to distribute these records to maximize concurrency. Several heuristic methods of distributing data for multikey search applications have been proposed. Most of them have been developed for partial match retrieval type queries. Modulo based distribution methods have been proposed in Refs 2 and 14, and Fieldwise eXclusive-or (FX) distribution methods have been proposed in Ref. 5. Both methods give sufficient conditions for optimal distribution. However, in all these methods, range specification in a query is not considered.

In this paper we use a partitioned hash directory for the access structure of a file. This directory is based on multikey hashing schemes,1 where each hash function is order-preserving. Multikey hash function, \(H\), for a file consisting of \(n\) fields is an ordered \(n\) functions \(h_1, \ldots, h_n\) such that given a record \(r = \langle r_1, \ldots, r_n \rangle\), \(H(r) = \langle h_1(r_1), \ldots, h_n(r_n) \rangle\). \(H(r)\) is called a bucket. Several order-preserving hash functions have been proposed in Ref. 3. A method for implementing order-preserving hash accesses can also be found in Ref. 9. The objective of this paper is to investigate optimal bucket distribution for range queries in multikey hashing.

The remainder of this paper is organized as follows. Section 2 describes basic notations as well as relevant definitions and assumptions. In section 3 we show the inherent limitations for optimal data distribution for some types of range queries. Section 4 and 5 describe optimal data distribution methods for range queries. The optimality conditions for these methods are also given. Section 6 contains the concluding remarks.

2. DEFINITIONS AND TERMINOLOGY

In this section we describe basic notations and definitions.

**Definition**

- \(f = \{0, 1, \ldots, F_i - 1\}\), a set of hashed values of field \(i\) by the \(i\)-th hashing function in multikey hashing.
- \(F_i\) denotes \(\{f_i\}\).
- \(M\) denotes the number of parallel devices.
- \(I: N \rightarrow N\) denotes the identity function, where \(N\) is the set of all natural numbers including 0.
- \(Z_M\) is the set of all integers from 0 to \(M-1\).
- \(r(u, v)\) denotes the range specification between \(u\) and \(v\), where both boundaries are closed.

We assume that a query can have three types of field value specification which are single value, range, and don’t care in which case it is denoted by \(*\). We will not use \(r(u, v)\) to denote a single value (i.e., range size is equal to 1).
to one) and don’t care (i.e. unspecified). We also do not allow cyclic ranges like \( r(S, 2) \), i.e. \( 5, 6, \ldots, F(t)-1, 0, 1, 2 \). In other words, whenever \( r(u, v) \) is a range specification for field \( i \), it is implied that \( u < v \) and, either \( u = 0 \) or \( v = F(t)-1 \).

**Definition.** A range query for files with \( n \) fields is denoted by \([A_1, A_2, \ldots, A_n]\), where for each \( i = 1, \ldots, n \), \( A_i = * \), or \( r(u, v) \), or \( w \), where \( u, v, w \in f_i \) and \( u < v \).

**Definition.** When the number of range specified fields is \( \alpha \) in a query \( q \), query \( q \) is called type \( \alpha \) range query.

**Example 1.** Let \( q_1, q_2 \) and \( q_3 \) be queries in the file (DEPT, AGE, STATE) such that \( q_1 = \text{[Math, r(20, 27), Ohio]} \), \( q_2 = \text{[* r(0, c - 1), *]} \) and \( q_3 = \text{[r(l, c), *]} \). Let \( S_{1} = \{s_1, s_2, s_3\} \) and \( M = 4 \). Since the sets \( M(s_1 - 1, T), \ldots, M(s_3 - 1, T) \) are also mutually disjoint, and the size of cartesian product of these sets is \( M \), \( M(s_1 - 1, T) = M(s_3, T) \). We will show a contradiction for either case when the size of \( T \) is even, or odd.

**Theorem 1.** Let \( T_1 = \{t_1, \ldots, t_{2^{\alpha}}\} \), and \( T_2 = \{t_{2^{\alpha}+1}, \ldots, t_n\} \). The sets \( M(S_1 - 1, T_1) \) and \( M(S_3, T_2) \) are disjoint (otherwise, the distribution is not strict optimal for query \( q_3 = \text{[r(s_1 - 1, s_2), r(t_1, t_2)]} \)). Thus, \( M(s_1 - 1, T_1) \) is equal to \( M(s_2, T_1) \), and \( M(s_3 - 1, T_2) \) is equal to \( M(s_3, T_1) \). Let us now consider range query \( q_6 = \text{[r(s_1 - 1, s_2), r(t_1, t_2)]} \). Suppose the bucket \( \langle s_1 - 1, t_{2^{\alpha}+1} \rangle \) is stored at device \( m_0 \) and the bucket \( \langle s_3, t_{2^{\alpha}+1} \rangle \) is stored at device \( m_1 \). Since \( t_{2^{\alpha}+1} \in T_2 \) and \( M(s_1 - 1, T_1) \cap M(s_3, T_1) \) is not an element of the devices at which bucket \( \langle s_3, t_{2^{\alpha}+1} \rangle \) is stored. However, the devices in

\[
\left\{ \bigcup_{k=1}^{\alpha-1} M(s_k, t_{2^{\alpha}+2}, \ldots, t_n) \right\}
\]

do not have any qualified bucket for query \( q_3 \). This contradicts the assumption.

**Theorem 2.** When a file consists of \( n (n \geq 3) \) fields and the given number of devices is \( M \), there does not exist a data distribution which is strict optimal for any type 0 range query that has at most two unspecified fields, and for any type 1 range query that has at most one unspecified field if (i) there are at least three fields \( i, j \) and \( k \) such that \( F_k \geq F_j \geq F_i \) and \( M < F_i F_k \leq 2M \), and \( F_i \geq F_j + F_k / M \), and (ii) there are two integers \( i \) and \( j \) such that \( bf_i = cf_j \).  

**proof** It is sufficient to prove the theorem for files with only three fields. Suppose the theorem is not true. Then, we have a distribution which is strict optimal for any query in the theorem. Let \( cF_i = M \) and \( bF_i = M \). This implies that \( c < F_i \) and \( b < F_i \). Suppose bucket \( \langle 0, 0, 0 \rangle \) is stored at device \( m_0 \). Bucket \( \langle 0, 0, 1 \rangle \) is stored at \( m_1 \), and bucket \( \langle 0, 0, F_i - 1 \rangle \) is stored at \( m_{n-1} \).

Clearly, \( m_0, \ldots, m_{n-1} \) are all different. We will show that buckets \( \langle 0, c, b \rangle \) and \( \langle c, 0, b \rangle \) should be stored at device \( m_0 \). At first, \( M(0, 0, f) \) is equal to \( M(0, c, f) \) (otherwise, the distribution is not strict optimal for at least one of the range queries \( q_2 = \text{[r(0, c - 1), *]} \) and \( q_3 = \text{[r(1, c), *]} \)). Thus, the possible set of devices at which bucket \( \langle 0, c, b \rangle \) can be stored is \( \{m_0, \ldots, m_{n-1}\} \). Bucket \( \langle 0, c, b \rangle \) cannot be stored at any one of the devices \( m_1, \ldots, m_n \). Otherwise, for range query \( q_5 = \text{[0, *, r(1, b)]} \), at least one of the devices \( m_1, \ldots, m_n \) has two or more qualified buckets while there exist a device which does not have any qualified bucket. Bucket \( \langle 0, c, b \rangle \) cannot also be stored at any one of the devices \( m_{n+1}, \ldots, m_{n+1} \). Otherwise, the distribution is not strict optimal for range query \( q_4 = \text{[r(0, c - 1), 0, *]} \) and \( q_5 = \text{[r(1, c), 0, *]} \). The bucket \( \langle 0, c, b \rangle \) should be stored at device \( m_0 \). Now, in order for the distribution to be strict optimal for queries \( q_2 = \text{[r(0, c - 1), 0, *]} \) and \( q_5 = \text{[r(1, c), 0, *]} \), \( M(c, 0, f) \) should be the same as \( M(0, 0, f) \). Thus, the possible set of devices at which bucket \( \langle c, 0, b \rangle \) can
be stored is \( \{m_1, \ldots, m_F\} \). Bucket \((c,0,b)\) cannot be stored at any one of the devices \( m_1, \ldots, m_F \). Otherwise, the distribution is not strict optimal for at least one of the queries \( q_1 = [*,0, r(1,b)] \) and \( q_2 = [*,0, r(b,F-1)] \). Thus, bucket \((c,0,b)\) should be stored at device \( m_q \). However, this bucket distribution is not strict optimal for query \( q_2 = [*,0, r(0,c,b)] \) because device \( m_q \) has at least two qualified buckets for \( q_2 \) while there exists a device which does not have any qualified bucket for \( q_2 \). This is a contradiction.

It has been shown in Ref. 5 that when each field size and the given number of devices are power of 2, perfect optimal distribution for type 0 range queries is always possible if the number of fields whose sizes are less than the given number of devices, is no greater than four. The data distribution methods for type 0 range queries are also given in that paper. In Ref. 11 it has been shown that perfect optimal distribution does not exist for binary cartesian product files with \( n \) fields, if \( M \geq 4 \) and \( n = \lceil \log_2 M \rceil + 2 \). This implies that perfect optimal distribution for type 0 range queries is not always possible for files with four or more fields.

We have shown that there are inherent limitations to achieve perfect optimal distribution for range queries even for a file consisting of small number of fields. In the following sections we present Fieldwise eXclusive-or (FX) distribution methods for range queries. These methods were first investigated in Ref. 5, where objective applications are type 0 range queries (i.e., partial match queries). In order to present FX distribution methods for range queries it is necessary to describe definitions and terminology for FX distribution methods.

### 4. Definitions and Terminology for FX Distribution Methods

This section gives definitions and terminology for FX distribution methods. For convenience we assume, from now on, that each field size and the given number of devices \( M \) are power of 2.

**Definition.** \( (a_{m-1} \ldots a_0)_2 \) is a binary notation of an integer, where \( a_i \) is a binary digit.

**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 \( X = (a_{m-1} \ldots a_0)_2 \) and \( Y = (b_{m-1} \ldots b_0)_2 \) are two integers, \( X[+Y] = (a_{m-1}[+b_{m-1}] \ldots a_0[+b_0])_2 \). 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 \( \{x_1[+y_i] \mid x_i \in X, y_i \in Y\} \).

#### 4.1 The Basic FX Distribution Method

Let \( f_1 \times f_2 \times \ldots \times f_n \) be a set of all buckets. The Basic FX distribution method allocates bucket \( \langle J_1, \ldots, J_n \rangle \) into device \( T_{\mathfrak{m}}(\{+[J]\}_{0\leq i \leq n}(J)) \), where \( T_{\mathfrak{m}} : N \rightarrow Z_M \) is a function which returns only the rightmost \( \log_2 M \) bits of domain values, and \( J \in f_i \) for \( j = 1, \ldots, n \). Note that \( \{+[J]\}_{0\leq i \leq n}(J) \) is a shorthand notation for performing exclusive-or operation between sets of integers \( J_1, \ldots, J_n \).

**Example 2.** Table 1 shows the bucket distribution by the Basic FX distribution method, where \( f_3 = \{0,1,2,3\} \) and \( M = 4 \). In this figure, binary numbers are used for field values and decimal numbers are used for Device No. We can verify that the distribution in Table 1 is perfect optimal for type (0–1) range queries but is not perfect optimal for type (0–2) range queries. In fact, by Theorem 1 there does not exist a perfect optimal distribution for type (0–2) range queries for this file system.


<table>
<thead>
<tr>
<th>( f_1 )</th>
<th>( f_2 )</th>
<th>Device No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>000</td>
<td>001</td>
<td>1</td>
</tr>
<tr>
<td>000</td>
<td>010</td>
<td>2</td>
</tr>
<tr>
<td>000</td>
<td>011</td>
<td>3</td>
</tr>
<tr>
<td>001</td>
<td>000</td>
<td>1</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>010</td>
<td>3</td>
</tr>
<tr>
<td>001</td>
<td>011</td>
<td>2</td>
</tr>
<tr>
<td>010</td>
<td>000</td>
<td>2</td>
</tr>
<tr>
<td>010</td>
<td>001</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>010</td>
<td>0</td>
</tr>
<tr>
<td>010</td>
<td>011</td>
<td>1</td>
</tr>
<tr>
<td>011</td>
<td>000</td>
<td>3</td>
</tr>
<tr>
<td>011</td>
<td>001</td>
<td>2</td>
</tr>
<tr>
<td>011</td>
<td>010</td>
<td>1</td>
</tr>
<tr>
<td>011</td>
<td>011</td>
<td>0</td>
</tr>
</tbody>
</table>

#### 4.2 Extended FX Distribution Methods

Extended FX distribution methods are defined by using the same principle of the Basic FX distribution method along with various field transformation techniques. Let \( f_1 \times f_2 \times \ldots \times f_n \) be a set of all buckets. Extended FX distribution methods allocate bucket \( \langle J_1, \ldots, J_n \rangle \), \( J \in f_i \) for \( j = 1, \ldots, n \), into device \( T_{\mathfrak{m}}(\{+[J]\}_{0\leq i \leq n}(X(j))) \), where:

(i) if \( |J| \geq M \), \( X_j \) is the identity function,

(ii) if \( |J| < M \), \( X_j \) is an element of the set of injective (one-to-one) functions whose domains are \( f_i \) and ranges are \( Z_M \).

\( X_j \) is called a field transformation function.

When \( X_j \) is the identity function for all \( j = 1, \ldots, n \), Extended FX distribution methods reduce to the Basic FX distribution method. Thus, the Basic FX distribution method is a special case of Extended FX distribution methods. From now on, we will call FX distribution methods instead of Extended FX distribution methods.

Several field transformation functions have been proposed and shown to be effective for type 0 range queries in Ref. 5. Thus, in this paper we will not discuss the cases when only type 0 range queries are involved.

### 5. FX Distribution Methods for Range Queries

In section 5.1 we describe the conditions for optimal distribution for various types of range queries when the Basic FX distribution method is used. In section 5.2 we will present field transformation functions developed for range queries. Following facts have been proved in Refs 5 and 10, and will be used frequently in proving theorems in this section.

**Fact 1.** For any type 0 range query which has at least one unspecified field whose size is greater than or equal

---

**THE COMPUTER JOURNAL, VOL. 35, NO. 1, 1992 47**
to \( M \), all the devices have equal number of qualified buckets by the Basic FX distribution method.

**Fact 2.** The Basic FX distribution method is strict optimal for any type 0 range query which has at most one unspecified field.

**Fact 3.** For a nonnegative integer \( L \), let \( L = aw + b \), where \( w \) is a power of 2, and \( a, b \) are some nonnegative integers such that \( 0 \leq b \leq w - 1 \). When \( Z_w = \{0, 1, \ldots, w-1\}, Z_{aw+b} = \langle aw, aw+1, \ldots, (a+1)w - 1 \rangle \).

### 5.1 Optimal distribution for range queries by the Basic FX distribution method

This section gives conditions for optimal distribution for range queries by using the Basic FX distribution method.

**Theorem 3.** The Basic FX distribution method is strict optimal for any range query, for which there exists at least one range specified field whose size is greater than or equal to the given number of devices \( M \).

**<proof>** Since any range query is a set of type 0 range queries, and by Fact 1 we know that equal number of qualified buckets are distributed in all the devices for each type 0 range query, the proof immediately follows.

**Corollary 3.1.** The Basic FX distribution method is strict optimal for any range query, for which there exists at least one range specified field such that the specified range size is an integral multiple of the given number of devices \( M \).

**Theorem 4.** The Basic FX distribution method is strict optimal for any range query, for which there is at most one range specified field and all the other fields are specified as single values.

The proof is almost the same as that of Fact 2 which is given in Ref. 5.

**Theorem 5.** When all the field sizes are greater than or equal to the given number of devices \( M \), the Basic FX distribution method is strict optimal for type (0–1) range queries.

**<proof>** When all the fields are specified as a single value, the proof is trivial. When at least one field is unspecified, the proof follows by Theorem 3. The only remaining case is when one field is specified as a range and all the other fields are specified as single values. The proof for this case follows by Theorem 4.

Theorems 3, 4, and 5 show general characteristics of exclusive-or operation for optimal distribution. However, the Basic FX distribution method does not give optimal distribution for many types of range queries. The following proposition gives the conditions for optimal distribution for these cases.

**Proposition 1.** Let \( q(f) = \{i_1, i_2, \ldots, i_k\} \) be the set of range specified or unspecified fields for a range query \( q \). Let \( S_{i_j} \) be the set of elements in the range when field \( i_j \) is range specified, or be \( f_{i_j} \) when field \( i_j \) is unspecified. Then, FX distribution methods are strict optimal for a range query \( q \), if there exists a set of fields \( \{i_1, \ldots, i_k\} \subseteq q(f) \) such that \( |S_{i_1} \times \ldots \times S_{i_k}| = \text{an integral multiple of} \ M \), and \#\( \{i_1, \ldots, i_k\} \subseteq S_{i_1} \times \ldots \times S_{i_k} | T_q([\lesssim i_{i_k}]) = z \) for all \( z \in Z_M \).

The proof for type 0 range queries is given in Refs 5 and 10. The proof of the proposition for type 1 or more range queries is straightforward from that of type 0 range queries.

The proposition says that optimal distribution for a subset of fields gives strict optimal distribution for many range queries in which those fields are unspecified or range specified. However, when the size of none of the fields is greater than or equal to \( M \), the conditions given in Proposition 1 are not satisfied in the Basic FX distribution method. Thus, in the next section we propose field transformation techniques for the fields whose sizes are less than the given number of devices \( M \). By the definition of field transformation function, it is easy to see that all the theorems that hold for the Basic FX distribution method also hold for FX distribution methods.

### 5.2 Field transformation functions for range queries

In this section we present field transformation techniques which can improve the performance over the Basic FX distribution method significantly. The following paragraph exemplifies the idea.

When \( f_1 = \{0, 1, 2, 3\}, f_2 = \{0, 1, 2, 3\} \) and \( M = 8 \), the distribution by the Basic FX distribution method is not strict optimal for many type 0 and type 1 range queries. We can easily see that the distribution in Figure 1 is not strict optimal for many type 0 and type 1 range queries when \( M = 8 \). Suppose \( X \) is an oneto-one mapping such that \( X(0) = 0, X(1) = 4, X(2) = 2, X(3) = 6 \). When each bucket \( \langle u, v \rangle \) in the file \( f_1 \times f_2 \) is stored at the device \( u + X(v) \), the distribution is strict optimal for any range query in which field 1 is not specified as a range (we do not give an example but it can be easily verified). Thus, our objective is to find such mapping, \( X \) in general, for the given file system. The following proposition gives sufficient condition for the mapping \( X \).

**Proposition 2.** Let a file consist of two fields, \( i, j \) whose sizes are less than the given number of devices \( M \), and \( X \) be an injective function from \( N \) to \( Z_M \). Then, FX distribution method applying function \( X \) to field \( i \) is strict optimal for any range query in which field \( i \) is not range specified, if for any \( L \) consecutive values \( J_1, \ldots, J_L \) in field \( j \) such that \( L \leq M/F_i \), \( \{X(J_1), \ldots, X(J_L)\} = \{k_x F_i + c_x | x = 0, \ldots, L-1, \text{and for all } x, k_x \in N, k_x \leq M/F_i, 0 \leq c_x < F_i, \text{and } k_x + c_x \leq L \} \).

The proof for the proposition is straightforward from Fact 3. Based on Proposition 2, we describe an idea of how to find the mapping \( X \) in the following paragraph.

Let \( f_1 = 8 \) and \( M = 16 \). The sizes of field \( i \) in which we are interested are 2, 4, and 8. Note that when the size of field \( i \) is greater than or equal to \( M \), by Theorem 3 and Fact 2 the distribution is strict optimal for any range query in which field \( i \) is not range specified. When \( F_i = 8 \), in order to satisfy the condition of the mapping \( X \) in Proposition 2, \( X(0), X(1), \ldots, X(7) \) should be in intervals \( [0, 8) \) and \( [8, 16) \), alternately. Here, ‘\( ' \) and ‘\( ' \) denote ‘closed’ and ‘open’, respectively. When \( F_i = 8 \), \( X(0), X(1), \ldots, X(7) \) should be in intervals \( [0, 4), [4, 8), [8, 12) \) and \( [12, 16) \) in turn. Since the sequence of intervals also has to satisfy the case when \( F_i = 8 \), the above sequence of intervals has to be reordered such as \( [0, 4), [4, 8), [8, 12) \) and \( [12, 16) \) in turn. Similarly, we have to reorder the sequences of intervals such as \( [0, 2), [2, 4), [4, 6) \), \( [6, 8), [8, 10), [10, 12), [12, 14), [14, 16) \) in turn.

This idea can be stated as follows: when \( (\alpha_1, \ldots, \alpha_n)_b \) is the binary notation of \( J \in f_j \), where \( m = \log_2 M \);
OPTIMIZING DATABASE ACCESSES

(1) For any consecutive two \( f_i, f_j \in f_p \), the most significant bit values of \( X(f_i) \) and \( X(f_j) \) have to be different.

(2) For any consecutive four \( f_i, f_j, f_k, f_l \in f_p \), two most significant bits (i.e. leftmost two bits) of \( X(f_i), X(f_j), X(f_k), X(f_l) \) have to be all different.

(3) Thus, in general for any consecutive \( d \) values \( f_i, \ldots, f_j \in f_p \), where \( d \) is a power of 2, the \( \log_2 d \) most significant bits of \( X(f_i), \ldots, X(f_j) \) have to be all different.

Based on these observations we propose the following field transformation functions.

Definition: When \( (a_0, a_1, a_2, a_3) \) is the binary notation of \( \ell \in f_p \), \( UR^M; \ell \rightarrow Z_m \) is a function such that \( UR^M(\ell) = (a_0a_1 \ldots a_{m-2}a_{m-1})_m \) where \( m = \log_2 M \).

Definition: When \( |f| < M \), \( UM^{M,1/\ell}; \ell \rightarrow Z_M \) is a function such that \( UM^{M,1/\ell}(\ell) = UM^M(\ell)[+](\ell \mod d^{M,1/\ell}) \), where \( d^{M,1/\ell} = M/|f| \).

Example 3. When \( f_1 = \{0, 1, 2, 3\} \), \( f_2 = \{0, 1, 2, 3, 4, 5, 6, 7\} \) and \( M = 16 \),

(a) \( UR^{16}(f_1) = \{0, 8, 4, 12\} \) and \( UM^{16,4}(f_1) = \{0, 9, 6, 15\} \).

(b) \( UR^{16}(f_2) = \{0, 8, 4, 12, 2, 10, 6, 14\} \) and \( UM^{16,8}(f_2) = \{0, 9, 4, 13, 2, 11, 6, 15\} \).

Lemma 6.1. When there are only two fields \( i, j \) such that \( |f_i| \) is less than the given number of devices \( M \), functions \( UR^M \) and \( UM^M \) satisfy the condition of the function \( X \) in Proposition 2.

The proof of the lemma is straightforward from the definition of \( UR^M \) and \( UM^M \).

Because of notational complexity we will use the following conventions: When the parameter \( M \) of a function \( UR \) denotes the given number of devices, we will leave out the parameter \( M \) by default. The parameter \( M \) and \( |f| \) of function \( UR \) will be also left out whenever there is no ambiguity.

Proposition 2 is only for the case when field \( i \) is \( I \)-transformed (note that \( I \) denotes the identity function). In Proposition 3 we generalize Proposition 2 to include cases when other transformation functions are applied to field \( i \).

Proposition 3. When a file consists of two fields \( i, j \) whose sizes are less than the given number of devices \( M \), let \( X_i \) and \( X_j \) be transformation functions applied to field \( i \) and \( j \), respectively. Let \( d_i = M/E_i \). Then, FX distribution methods with \( X_i(f_i) \) and \( X_j(f_j) \) are strict optimal for any range query in which field \( i \) is not range specified if

(1) \( X_i(f_i)[+]X_j(f_j) \) and \( X_i(f_j)[+]X_j(f_i) \) are disjoint for any \( J, K \in f_i \) such that \( J[+]< d_i \) and

(2) \( X_i(f_i)[+]X_j(f_j) = X_i(f_j)[+]X_j(f_i) \) for any \( J \in f_j \) and for any \( a \in N \) such that \( J[+a]< d_i \).

\(<proof> \) (1) and (2) in the proposition guarantee that for any \( J \in f_j \), the sets \( M(f_i,J), M(f_i,K+1), \ldots, M(f_i, J+d_i-1) \) are disjoint if \( J+d_i-1 < E_i-1 \), and \( M(f_i,J) = M(f_i,J+a) \) for any \( a \in N \) such that \( J+a < E_i-1 \).

Thus, the proof follows. \( \blacksquare \)

Proposition 3 will be used frequently in proving theorems.

We will show through Theorem 6 through Theorem 9 that FX distribution methods with the proposed field transformation functions give optimal distribution for several types of range queries.

Definition. Let \( S = \{s_0, \ldots, s_k\} \) be a set of nonnegative integers. Then for any nonnegative integer \( c \) we define \( S+c = \{s_0+c, \ldots, s_k+c\} \).

Lemma 6.2. Let \( f_i = \{0, 1, \ldots, E_i-1\} \) such that \( F_i < M \).

Let \( d_i = M/E_i \). Then, for any nonnegative integer \( J \) and \( c \) such that \( 0 \leq J < F_i \) and \( 0 \leq c < d_i \), \( UR(f_j)[+](Jd_i+c) = UR(f_j)+c \).

The proof of the lemma is given in Ref. 9.

Theorem 6. When there are only two fields \( i, j \) such that \( F_i \) is less than the given number of devices \( M \), the FX distribution method with \( I \)-transformation for field \( i \) and \( UR \)-transformation for field \( j \) is (i) perfect optimal for type (0–1) range queries when \( F_j > M \), and (ii) perfect optimal for type (0–2) range queries when \( F_i > F_j \).

\(<proof> \) When one field is range specified and the other field is unspecified, the proof follows from Theorem 4. When field \( i \) is range specified and field \( j \) is unspecified, the proof follows from Lemma 6.2. When field \( j \) is range specified and field \( i \) is unspecified, the proof follows by Lemma 6.1 and Proposition 2. When there is no range specification, the proof is given in Ref. 5.

Example 4. Let \( f_1 = \{0, 1, 2, 3\} \), \( f_2 = \{0, 1, 2, 3\} \) and \( M = 8 \). Table 2 shows the bucket distribution by the FX distribution method with \( I(f_1) \) and \( UR(f_2) \), where \( UR(f_2) = \{0, 4, 2, 6\} \). We can verify that the distribution in Table 2 is strict optimal for any type 0 and type 1 range query.

<table>
<thead>
<tr>
<th>( R(f_i) )</th>
<th>( UR(f_j) )</th>
<th>Device No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>000</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>000</td>
<td>010</td>
<td>2</td>
</tr>
<tr>
<td>000</td>
<td>010</td>
<td>2</td>
</tr>
<tr>
<td>001</td>
<td>000</td>
<td>1</td>
</tr>
<tr>
<td>001</td>
<td>100</td>
<td>5</td>
</tr>
<tr>
<td>001</td>
<td>010</td>
<td>3</td>
</tr>
<tr>
<td>001</td>
<td>110</td>
<td>7</td>
</tr>
<tr>
<td>010</td>
<td>000</td>
<td>2</td>
</tr>
<tr>
<td>010</td>
<td>100</td>
<td>6</td>
</tr>
<tr>
<td>010</td>
<td>010</td>
<td>2</td>
</tr>
<tr>
<td>010</td>
<td>110</td>
<td>4</td>
</tr>
<tr>
<td>011</td>
<td>000</td>
<td>3</td>
</tr>
<tr>
<td>011</td>
<td>100</td>
<td>7</td>
</tr>
<tr>
<td>011</td>
<td>010</td>
<td>1</td>
</tr>
<tr>
<td>011</td>
<td>110</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 2. FX distribution with \( I(f_1) \) and \( UR(f_2) \)

Now, it is worth considering the following question: Do there exist three field transformation functions \( X_i \), \( X_j \) and \( X_k \) such that for any two fields whose sizes are less than \( M \), FX distribution methods with any two of \( X_i \), \( X_j \), \( X_k \) transformation functions are perfect optimal for type (0–1) range queries? Unfortunately, there do not exist such transformation functions because otherwise, it immediately contradicts Theorem 2. This implies that it is inevitable to have some restrictions for perfect optimal distribution for type (0–1) range queries when \( I \) and \( UM \)-transformation, and \( UR \) and \( UM \)-transformation are considered.

Lemma 7.1. When there are only two fields \( i \) and \( k \) whose sizes are less than the given number of devices \( M \), the FX distribution method with \( I \)-transformation for field \( i \) and \( UM \)-transformation for field \( k \) is strict optimal for any range query in which field \( i \) is not range specified.
whose sizes are less than the given number of devices $M$ and $F_i \leq F_k$, the FX distribution method with $I$-transformation for field $i$ and $UM$-transformation for field $k$ is strict optimal for any range query in which field $k$ is not range specified.

**Proof** Let $d_a = M/F_k$. It is not difficult to see that for any two different $K_i$ and $K_j$ which are in the same interval of size $d_a$, $UM(f_i) [+] K_i$ and $UM(f_j) [+] K_j$ are disjoint. Thus, by Proposition 3 it is sufficient to show that for any $J \in f_i$, $UM(f_j) [+] J = UM(f_j) [+] (J [+] ad_k)$, where $\alpha$ is any nonnegative integer such that $ad_k < F$. Now,

$$UM(f_j) [+] ad_k = UM(f_j) [+] UR(M(UR^a(\alpha))) = UM(f_j) [+] (UR^a(\alpha) \mod d_a)$$

The first equality holds because $UR^M(UR^a(\alpha)) = UR^8(UR^a(\alpha))d_a$, and $UR^8(UR^a(\alpha)) = \alpha$. The second equality holds because for any $K \in f_i$, $UM(f_j) [+] (K \mod d_a) = UM(f_j) [+] (K/ \mod d_a)$. Since $\alpha < F$, it implies that $UR^a(\alpha)$ is a multiple of $d_a$ for all $\alpha = 0, \ldots, d_a - 1$. Thus, $UR^a(\alpha) \mod d_a = 0$ and therefore the proof follows.

**Theorem 7** When there are only two fields $i$ and $k$ whose sizes are less than the given number of devices $M$, the FX distribution method with $I$-transformation for field $i$ and $UM$-transformation for field $k$ is (i) strict optimal for any range query in which field $i$ is not range specified if $F_i > F_k$, and (ii) perfect optimal for type $(0-1)$ range queries if $F_i \leq F_k$, and (iii) perfect optimal for type $(0-2)$ range queries if $F_i F_k \leq M$.

The theorem is a direct consequence of Lemma 7.1 and Lemma 7.2.

**Theorem 8** When there are only two fields $j$ and $k$ whose sizes are less than the given number of devices $M$, the FX distribution method with $I$-transformation for field $j$ and $UM$-transformation for field $k$ is (i) strict optimal for any range query in which field $k$ is not range specified if $F_j \leq F_k$, and (ii) strict optimal for any range query in which field $j$ is not range specified if $F_j \geq F_k$, and (iii) perfect optimal for type $(0-1)$ range queries if $F_j = F_k$, and (iv) perfect optimal for type $(0-2)$ range queries if $F_j F_k \leq M$.

**Proof** We give the proof for (i) only. Proof for other cases are straightforward based on that of (i). Complete proof can be found in [Ref. 9]. Let $d_a = M/F_k$. Since $UM(f_j) [+] J$ and $UM(f_j) [+] J$ are disjoint, by Proposition 3 it is sufficient to show that for any $J \in f_j$, $UM(f_j) [+] (J \mod d_a) = UM(f_j) [+] UR(J [+] ad_k)$, where $\alpha$ is any nonnegative integer such that $ad_k < F$. Since $UM(f_j) [+] (J \mod d_a) = UM(f_j) [+] (J \mod d_a)$ for any $J \in f_j$, this implies that $UM(f_j) [+] UR(J [+] ad_k) = UM(f_j) [+] (J [+] ad_k \mod d_a) = UM(f_j) [+] (J \mod d_a)$. Thus, the proof follows.

**Example 5** Let $f_i = \{0, 1, 2, 3\}$, $f_j = \{0, 1, 2, 3\}$ and $M = 8$. Table 3 shows the bucket distribution by FX distribution method with $UR(f_i)$ and $UM(f_j)$. Here, $UR(f_i) = \{0, 4, 2, 6\}$, $UM(f_j) = \{0, 5, 2, 7\}$ and Device No $= T_M(UR(f_i)) [+] UM(f_j)$, $J \in f_i J \in f_j$.

We can verify that the distribution is perfect optimal for type $(0-1)$ range queries.

**Theorem 9** When there are only three fields $i, j, k$ such that $F_i \leq F_k \leq F_j \leq M$, FX distribution methods can be always (i) perfect optimal for type 0 range queries, and (ii) strict optimal for any type 1 range query which has at most one unspecified field if $F_i = F_j$, or $F_i F_k \leq M$.

**Proof** The proof for (i) has been given in Ref. 5. Thus, let us consider only the case (ii). When $F_i \leq F_k$, let field $i$ be $I$-transformed, field $j$ be $UR$-transformed and field $k$ be $UM$-transformed. When $F_i F_k \leq M$, let field $i$ be $UR$-transformed, field $j$ be $I$-transformed and field $k$ be $UM$-transformed. Then, for both cases, the theorem is a direct consequence of Theorem 6, Theorem 7 and Theorem 8.

Note that by Theorem 2 there does not exist a data distribution method which guarantees strict optimal distribution for all the range queries in Theorem 9 (i.e., all possible type 0 range queries and all the type 1 range queries which have at most one unspecified field) for every file with three fields.

We have shown that FX distribution methods along with various combinations of field transformation functions give optimal distribution for many types of range queries. Here, it should be emphasized that the scope of optimality is increased significantly by these field transformation techniques along with Proposition 1. This is because by Proposition 1 optimal distribution for a subset of the fields guarantees strict optimal distribution for many range queries in which those fields are range specified or unspecified. Though FX distribution methods presented in this paper do not give optimal distribution in all the cases, these methods are strict optimal for a large class of range queries.

6. CONCLUSION

We have investigated file distribution problems for parallel processing of multiattribute range queries. Optimal distribution methods as well as the existence and nonexistence of perfect optimality have been presented. We have shown that perfect optimal distribution for multiattribute range queries does not exist in many file systems. We have given sufficient conditions for the nonexistence of perfect optimal distribution for various types of range queries. We have also presented optimal data distribution methods, called FX distribution, for several useful types of range queries. These methods are based on exclusive-or operation along

<table>
<thead>
<tr>
<th>Device No.</th>
<th>$UR(f_i)$</th>
<th>$UM(f_j)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>1</td>
<td>010</td>
<td>010</td>
</tr>
<tr>
<td>2</td>
<td>011</td>
<td>011</td>
</tr>
<tr>
<td>3</td>
<td>010</td>
<td>010</td>
</tr>
<tr>
<td>4</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>5</td>
<td>011</td>
<td>011</td>
</tr>
<tr>
<td>6</td>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>7</td>
<td>000</td>
<td>000</td>
</tr>
</tbody>
</table>
with various field transformation techniques. We have shown that FX distribution methods are perfect optimal for certain types of multiattribute range queries, and strict optimal for a large class of multiattribute range queries. However, there are many cases for which the existence of perfect optimal distribution is not known. We are currently investigating the existence of optimal distribution for these cases by extending FX distribution methods. This will require the development of more general transformation functions.

7. REFERENCES


Announcement

Switzerland (near Geneva), 28–30 October 1992

Third Eurographics Workshop on Object-Oriented Graphics

Call for Contributions

Aims and Scope

At past workshops it has been noted that people in the graphics community are addressing a wide variety of problems: surface modeling, rendering, animation, interaction, multiple media, constraints, etc. At the same time, it has also been noted that people share similar concerns in the areas of software reuse, extensibility and maintenance. This workshop will be an effort to identify such common concerns and to devise solutions that benefit a wide spectrum of research domains within the interactive graphics community. The goal of the workshop is to outline a common platform, based on a set of object-oriented primitives, for the support of graphics applications. To this end, each submission should be motivated by at least one of the following questions:

1. If my programming environment were to provide an object-oriented platform, what kind of support would I ask for it for my particular domain of interest?
2. As a designer of an object-oriented platform, how would it support object-oriented graphics?

Here, the term object-oriented platform should be taken in a broad sense; it may encompass any kind of functionality to support graphics applications, even non-graphic primitives such as support for persistent objects, concurrency, etc. The following non-exhaustive list provides some typical issues that could be discussed:

- support of reuse and extensibility in graphics systems;
- concurrency and distribution support for graphics applications;
- standards for data exchange such as geometric primitives, scripts, sound and video;
- the integration of constraints in graphics systems;
- the binding of input/output activities to objects and user interaction;
- encapsulation as a mechanism for interoperability of graphics applications;
- object-oriented 'wrapping' of existing graphics functions.

Full Papers

Please submit 4 copies of a paper (10–20 single-sided pages, 200 word abstract) for review by the programme committee. Accepted papers will be reproduced in the workshop proceedings. Invitations to submit revised versions for a book (in the Eurographic Seminars Series of Springer) will depend on the quality of the contributions. Shorter position papers (1–2 pages) may be accepted according to their relevance to the workshop. Questions concerning the suitability of papers should be addressed to the organizers.

Workshop Format

The workshop will be limited to 60 participants. To encourage discussion, at least two types of presentations are foreseen: 1) full paper presentations and 2) presentations of results from small group discussions. Equipment for document and transparency preparation will be made available. The results of the workshop may also be reproduced in the book if the participants and editors feel they are appropriate.

Schedule

31 May 1992 Deadline for paper submission
15 August 1992 Notification of acceptance of papers
28–30 October 1992 Workshop
15 January 1993 Deadline for revised papers

Venue and Fee

The workshop will be held in Switzerland. The fee will be about SFr. 700 including accommodation and meals. Limited funds for subsidizing students may be available.

Organization

The workshop is organized by Vicki de Mey and Xavier Pintado of the Centre Universitaire d’Informatique (University of Geneva) and promoted by Eurographics. The book will be edited by Wm Leler of Ithaca Software, Vicki de Mey and Xavier Pintado.

Address

Eurographics Workshop on Object-Oriented Graphics, Centre Universitaire d’Informatique, 12 rue du Lac, CH-1207 Geneva, Switzerland.
Tel.: 41 (22) 787 65 86; Fax 41 (22) 735 39 05.
E-mail: eeog@cui.unige.ch.