Chapter 3

Geographical Data Broadcast Cost Models

As discussed in Section 2.1, AT is further divided into two components, namely Probe Wait and Bcast Wait. We argue that it might be more appropriate to divide AT into four components: Index-Probe Wait (IPW), Index-Bcast Wait (IBW), Data-Probe Wait (DPW) and Data-Bcast Wait (DBW). IPW is the same as Probe Wait defined in (Imielinski, 1997), i.e., the time duration of getting to the nearest index segment. IBW is the time duration from the time when the first index segment is reached to the time when the last index segment is reached. DPW is defined as the duration from the time the last index segment is reached to the time when the first data segment is reached. DBW is defined as the duration from the time when the first data segment is reached to the time when the last data item is downloaded. The summation of IBW, DPW and DBW is equivalent to the Bcast Wait defined in (Imielinski, 1997). These four components are illustrated in Fig. 3-1 for two scenarios. We assume each index segment contains a certain number of pointers each of which points to a data segment, and each data segment contains only one data item. We use the intervals between the beginning and ending positions of data items as the measurements of the four components of access time as discussed in Chapter 2.
Fig. 3-1. The Four Components of Access Time

(a) Index and Data Are Multiplexed Into One Single Broadcast Channel

(b) Index and Data Use Two Separate Broadcast Channels

T₁: Time to begin accessing the broadcast channel
T₂: Time to reach the first requested index segment
T₃: Time when all requested index segments are downloaded
T₄: Time to reach the first requested data segment
T₅: Time when all requested data segments are downloaded
The definitions allow measuring the performance of index sequencing and data sequencing separately. Furthermore, they allow measuring the performance when index and data are broadcast using separate channels where the definitions in (Imielinski, 1997) only allow measuring the performance when index and data are broadcast using a single channel. In this study, we aim at reducing access time by providing a smarter arrangement of data items in the broadcast channel.

We believe reducing DBW is more important due to the following reasons. First, a client is able to remain in sleep mode during Probe Waits (IPW or DPW). However, it must switch modes during Broadcast Waits (IBW or DBW). Generally we can assume that the shorter the DBW is, the fewer mode switches occur. The reason is that data segments that need to be downloaded are more likely to be in a consecutive order for a smaller DBW, and thus, the number of mode switches among them can be reduced. Compared with IBW, DBW is much larger since data is usually much larger than its index, and thus, it is critical to minimize DBW. Second, only one pointer (the position of the nearest index segment) is recorded during IPW and at most $k$ pointers are recorded during IBW+DPW where $k$ is the number of data items in a query result set. However, there could be up to $k$ data items that are recorded during DBW. Since usually the size of a data item is much larger than that of a pointer, it is desirable to reduce DBW as much as possible to reduce energy consumption for storing data items that have already been downloaded during the data access process.
We are also interested in the total access time to data, i.e., \( AT_{Data} = DBW + DPW \). In this dissertation we consider two popular scenarios that involve both DBW and DPW. The first is that we assume the data and the index are multiplexed into one single broadcast channel. Here both the data and index segments are broadcast only once in a cycle and the index segments are placed ahead of the data segments. Without loss of generality we assume the index is placed at the beginning of a broadcast cycle since the broadcast sequence is cyclic. In this case, a client will have to wait for the beginning of a broadcast cycle (Fig. 3-1a) before accessing data items. The second scenario is that the data and index segments are broadcast in two separate channels and both the data and index segments are broadcast only once in their respective cycles. In this case, a client might begin to access the data channel at any position after it retrieves the pointers to the data segments from the index channel (Fig. 3-1b). We call the first scenario as “Multiplexing Scheme” or MUL, and the second scenario as “Separate Channels Scheme” or SEP.

Throughout this dissertation, we use “access time cost” to refer to the access time needed to complete a complex query over a broadcast sequence. For the rest of this chapter, we first propose the cost models to compute the DBW and the \( AT_{Data} \) under the two scenarios for a single complex query. We then present the cost models to compute the total DBW and the access time to data in the two scenarios for all queries over a data set. We handle spatial range queries for point data and network path queries for graph data separately although they both follow a similar framework.
3.1 **Cost Models for Processing a Single Complex Query**

Suppose the length of the broadcast cycle of a data channel is $L$. Let $L_2$ denote the access time of a single complex query result, i.e., DBW. Let $L_1$ and $L_3$ denote the time before $L_2$ and after $L_2$ in the broadcast sequence as shown in Fig. 3-2. Note that $L = L_1 + L_2 + L_3$.

![Diagram showing $T_1$, $T_2$, $T_3$, and $T_4$ with $L_1$, $L_2$, and $L_3$.]

- $T_1$: Time to begin a broadcast cycle
- $T_2$: Time to access the first required data item
- $T_3$: Time to access the last required data item
- $T_4$: Time to end a broadcast cycle

**Fig. 3-2. Illustration of $L_1$, $L_2$ and $L_3$**

We next compute the average cost for a single complex query under the Multiplexing scheme and Separate Channels scheme assuming a client begins to access the data channel randomly.

- **Multiplexing Scheme**: the access time cost to the data channel can be calculated as:

  $AT_{Data}^{Mul} = L_1 + L_2$

- **Using Separate Channels**: there are three cases that a client might begin to access the data channel. We compute their total access time costs separately and then compute the average access time cost.
Case 1: a client begins to access the data channel during $L_1$. Suppose its initial access position is $i$ then it has to wait for an amount of time equivalent to $(L_1-i)$ before downloading data during $L_2$, thus the total costs over all possible $i$ is:

$$Cost_1 = \sum_{i=0}^{L_1-1} (L_1 - i + L_2) = \frac{L_1(L_1 + 1)}{2} + L_1 * L_2$$

Case 2: a client begins to access the data channel during $L_2$. Regardless of its initial access position, it has to wait for the whole broadcast cycle to retrieve all the data items. Thus the total cost is:

$$Cost_2 = L * L_2$$

Note that there is a slight overestimation here as shown in Fig. 3-3 (the shaded data items are the ones in the query result set). We need a part of $L_2$ in the current broadcast cycle ($L_{2c}$) and a part of $L_2$ in the next broadcast cycle ($L_{2n}$). However their total might be less than $L_2$ since the rest of $L_2$, i.e., $L_{2m}= L_2 - L_{2c} - L_{2n}$, is not required. Suppose there are $n$ data items in the query result set and they are evenly distributed among $L_2$, then the overestimation on average is the half of the interval between two required data items among $L_2$, i.e., $L_2/(2n)$. We omit this overestimation to make our result more concise and easy to use as can be seen in computing the average shortly.

Fig. 3-3. Illustration of the Overestimation in Case 2

T0: Time to begin access the broadcast channel
T1: The end of the current broadcast cycle
T2: The beginning of the next broadcast cycle (The same as T1)
T3: Time to access the last required item in the next broadcast cycle
Case 3: a client begins to access the data channel in $L_3$. It has to wait for the rest of the time in $L_3$ in the current broadcast cycle and $L_1+L_2$ in the next broadcast cycle. Thus the total cost is:

$$\text{Cost } 3 = \sum_{i=0}^{L_3 - 1} (L_3 - i + L_1 + L_2) = \sum_{i=0}^{L_3 - 1} (L - i) = L_3 \cdot L - \frac{L_3(L_3 - 1)}{2}$$

Since there are totally $L$ initial access positions, the average is:

$$AT_{\text{Data}}^{\text{Sep}} = \frac{1}{L}(\text{Cost 1} + \text{Cost 2} + \text{Cost 3})$$

$$= \frac{1}{L} \left[ \frac{L_1(L_1 + 1)}{2} + L_1 \cdot L_2 + L \cdot L_2 + L_3 \cdot L - \frac{L_3(L_3 - 1)}{2} \right]$$

$$= \frac{1}{L} \left[ \frac{(L_1 + L_3)(L_1 - L_3 + 1)}{2} + L_1 \cdot L_2 + L \cdot (L_2 + L_3) \right]$$

$$= \frac{1}{L} \left[ \frac{(L - L_2)(L_1 - L_3 + 1)}{2} + L_1 \cdot L_2 + L \cdot (L - L_1) \right]$$

$$= \frac{1}{L} \left[ L^2 - L_1 \cdot (L - L_2) + \frac{(L - L_2)(L_1 - L_3 + 1)}{2} \right]$$

$$= \frac{1}{L} \left[ L^2 - \frac{(L - L_2)(L_1 + L_3 - 1)}{2} \right]$$

$$= \frac{1}{L} \left[ L^2 - \frac{(L - L_2)(L - L_1 - 1)}{2} \right]$$

From the result we can see that the average access time to the data channel in the Separate Channels scheme is determined only by $L$ and $L_2$. Usually the number of data items in a query result is far fewer than the total number of data items in a whole broadcast cycle, thus it is reasonable to assume $L \cdot L_2 >> 1$. Under this assumption, the formula can be simplified as $AT_{\text{Data}}^{\text{Sep}} = \frac{L}{2} + L_2 - \frac{L_2^2}{2L}$.
To further investigate the relationship between the average access time to the data channel and \( L_2 \), we can rewrite the formula as follows:

\[
\begin{align*}
AT_{Data}^{Sep} &= \frac{1}{L} \left[ L^2 - \frac{(L - L_2)^2 - (L - L_2)}{2} \right] \\
&= \frac{1}{L} \left[ L^2 - \frac{(L - L_2)^2 - \frac{1}{4} (L - \frac{L_2}{2})^2}{2} \right] \\
&= \frac{1}{L} \left[ L^2 + \frac{1}{8} - \frac{(L - L_2 - \frac{1}{2})^2}{2} \right]
\end{align*}
\]

Since \( L_2 < L \), the average cost decreases monotonically as \( L_2 \) decreases.

Let function \( g(L_2) \) be \( g(L_2) = \frac{L}{2} + L_2 - \frac{L_2^2}{2L} \). We will use this in the following analysis.

### 3.2 Spatial Range Query for Point Data

In this section we first compute all possible query result sets and their weights by exploring spatial semantics of a point data set and then we develop the cost models for DBW and the two scenarios of \( AT_{Data} \) by summarizing the weighted access time of individual query results we have developed in the last section.

Let \( DS = [x_1, x_2] \times [y_1, y_2] \) be the data space that defines all the geographical data points. Let the range query window size be \((q_x, q_y)\). We define the Extended Region \( R_u \) of point \( P_u \) as the rectangle of size \((q_x, q_y)\) centered at \( P_u \). As shown in Fig. 3-4, the distribution of the centers of the query window regions of size \((q_x, q_y)\) that contain the data item \( P_u \) is the extended region of \( R_u \). Furthermore, from Fig. 3-5 we can see that the distribution of the centers of the query window of size \((q_x, q_y)\) that contain both the data items \( P_u \) and \( P_v \) is the intersection of their extended regions \( R_u \).
and $R_v$. This relationship can be extended to higher orders, up to the intersected region among all $n$ extended regions.

**Fig. 3-4. The Possible Distribution of Centers of Query Regions That Contain $P_u$**

**Fig. 3-5. The Possible Distribution of Centers of Query Regions That Contain Both $P_u$ and $P_v$ (Shaded Area)**

We assume that all the locations inside the study region are equally likely to be the users’ locations at the time they issue a spatial range query, i.e., the centers of query windows. The access frequencies of a subset of data points resulted from the spatial range location-dependent queries is proportional to the area of the distributions
of the centers of the query windows that contain the subset of data points. Thus, the
access frequency of such data points in the subset is proportional to the intersection
area of their extended regions (area). Assume the number of spatial range queries
requested in the studied area is a fixed number (M), let
\[
c = \frac{M}{(x_2 - x_1) \times (y_2 - y_1)}
\]
then the access frequency of the query result set (freq) is
\[
freq = c \times area.
\]
For the sake of simplicity we omit the constant factor c and only use area as the access frequency for
a query result set. Note that the access frequency of a subset S is no less than the
access frequency of another subset S’ if \( S \subseteq S' \) since the intersection area of the
extended regions of the points in S’ is a subset of the intersected area of the extended
regions of the points in S.

Let \( A_i \) be the area of \( R_i \), \( A_{i,j} \) be the intersection area of \( R_i \) and \( R_j \), \( \ldots \), \( A_{1,2\ldots n} \) be
the intersection area of \( R_1, R_2 \ldots R_n \). Let \( A_i~\text{be the part of } A_i \text{ that solely contains point} \)
\( P_i \), \( A_{i,j} \) be the part of \( A_{i,j} \) that solely contains points \( P_i \) and \( P_j \), \( \ldots \), \( A_{1,2\ldots n} \) be the part of
the intersection area of \( R_1, R_2 \ldots R_n \) that contains all \( n \) points. It is easy to see that we
have the following relations:

\[
\tilde{A}_i = A_i - \bigcup_{j=1, i \neq j}^n A_{i,j}
\]
\[
\tilde{A}_{i,j} = A_{i,j} - \bigcup_{k=1, i \neq j \neq k}^n A_{i,j,k}
\]
\[
\ldots
\]
\[
\tilde{A}_{1,2\ldots n} = A_{1,2\ldots n}
\]
Let function $\pi(u)$ map point $u$ to its position in the broadcast sequence. According to our previous definitions and assumptions, the DBW for a single query result set that contains $k$ data items $n_1, n_2 \ldots n_k$, which is the definition of $L_2$ (c.f. Fig. 3-2), is $\max(\pi(n_1), \pi(n_2), \ldots \pi(n_k)) - \min(\pi(n_1), \pi(n_2), \ldots \pi(n_k))$. Correspondingly, $L_1$ is $\min\{\pi(n_1), \pi(n_2), \ldots \pi(n_k)\}$.

The total DBW cost for a query window $(q_x, q_y)$ is the summation of the weighted DBW for all possible query result sets. For a query result set that contains only two points $i$ and $j$, the interval between them in the broadcast sequence is $|\pi(i) - \pi(j)|$. Its weight is $\tilde{A}_{i,j}$ and its weighted DBW is $\tilde{A}_{i,j}^{(q_x, q_y)} * |\pi(i) - \pi(j)|$. Note that $i$ and $j$ can be any two points the extended regions of which intersect with each other. Similarly the weighted DBW for a query result set that contain three points, $i, j$ and $k$, is $\tilde{A}_{i,j,k}^{(q_x, q_y)} * (\max(\pi(i), \pi(j), \pi(k)) - \min(\pi(i), \pi(j), \pi(k)))$ and so on. Thus the total DBW cost for all possible query result sets with query window of size $(q_x, q_y)$ can be written as follows:

$$
\text{Cost}^{(q_x, q_y)} = \sum_{1 \leq i < j \leq n} \tilde{A}_{i,j}^{(q_x, q_y)} * |\pi(i) - \pi(j)|
$$
$$
+ \sum_{1 \leq i < j \leq k \leq n} \tilde{A}_{i,j,k}^{(q_x, q_y)} * (\max(\pi(i), \pi(j), \pi(k)) - \min(\pi(i), \pi(j), \pi(k)))
$$
$$
+ \ldots
$$
$$
+ \tilde{A}_{1,2,\ldots,n}^{(q_x, q_y)} * (\max(\pi(1), \pi(2), \pi(n)) - \min(\pi(1), \pi(2), \pi(n)))
$$
The final total cost of DBW is the summation of DBW Cost\(^{(q, q_j)}\) over all possible query windows Q, i.e.,

\[
\text{Cost}^{\text{DBW}} = \sum_{(q_i, q_j) \in Q} \text{Cost}^{(q_i, q_j)}
\]

\[
= \sum_{(q_i, q_j) \in Q} \sum_{1 \leq i < j \leq n} \tilde{A}_{i,j}^{(q_i, q_j)} \cdot |\pi(i) - \pi(j)|
\]

\[
+ \sum_{(q_i, q_j) \in Q} \sum_{1 \leq i < j \leq n} \tilde{A}_{i,j,k}^{(q_i, q_j)} \cdot (\max(\pi(i), \pi(j), \pi(k)) - \min(\pi(i), \pi(j), \pi(k)))
\]

\[
+ ... \sum_{(q_i, q_j) \in Q} \tilde{A}_{1,2,...,n}^{(q_i, q_j)} \cdot (\max(\pi(1), \pi(2)..., \pi(n)) - \min(\pi(1), \pi(2)..., \pi(n)))
\]

Let

\[
w_i = \sum_{(q_i, q_j) \in Q} \tilde{A}_{i}^{(q_i, q_j)}
\]

\[
w_{i,j} = \sum_{(q_i, q_j) \in Q} \tilde{A}_{i,j}^{(q_i, q_j)}
\]

\[
w_{i,j,k} = \sum_{(q_i, q_j) \in Q} \tilde{A}_{i,j,k}^{(q_i, q_j)}
\]

\[
... \sum_{(q_i, q_j) \in Q} \tilde{A}_{1,2,...,n}^{(q_i, q_j)}
\]

Then

\[
DBW = \sum_{1 \leq i < j \leq n} w_{i,j} \cdot |\pi(i) - \pi(j)|
\]

\[
+ \sum_{1 \leq i < j \leq n} w_{i,j,k} \cdot (\max(\pi(i), \pi(j), \pi(k)) - \min(\pi(i), \pi(j), \pi(k)))
\]

\[
+ ... \sum_{1 \leq i < j \leq n} w_{1,2,...,n} \cdot (\max(\pi(1), \pi(2)..., \pi(n)) - \min(\pi(1), \pi(2)..., \pi(n)))
\]
Similarly, the costs of access time to data under the Multiplexing scheme, $AT_{Data}^{Mul}$, and Separate Channels scheme, $AT_{Data}^{Sep}$, are as follows:

$$AT_{Data}^{Mul} =$$
$$= \sum_{1 \leq i \leq n} w_i \pi(i)$$
$$+ \sum_{1 \leq i < j \leq n} w_{i,j} \max(\pi(i), \pi(j))$$
$$+ \sum_{1 \leq i < j < k \leq n} w_{i,j,k} \max(\pi(i), \pi(j), \pi(k))$$
$$+ \ldots$$
$$+ w_{1,2,\ldots,n} \max(\pi(1), \pi(2)\ldots \pi(n))$$

$$AT_{Data}^{Sep} =$$
$$\sum_{1 \leq i < j \leq n} w_{i,j} g(\pi(i) - \pi(j))$$
$$+ \sum_{1 \leq i < j < k \leq n} w_{i,j,k} g(\max(\pi(i), \pi(j), \pi(k)) - \min(\pi(i), \pi(j), \pi(k)))$$
$$+ \ldots$$
$$+ w_{1,2,\ldots,n} g(\max(\pi(1), \pi(2)\ldots \pi(n)) - \min(\pi(1), \pi(2)\ldots \pi(n)))$$

Note that we omit the access times of queries that only have a single data item (which is $L/2$) in the SEP scheme since they are constant and do not contribute to the determination of optimal ordering.
3.3 Network Path Query for Graph Data

Let $V$ denote the vertex set of the network. Let $S_{ij}$ denote a path sequence with access frequency $f(i,j)$ for source vertex $i$ and destination vertex $j$. Assume the order of the $k$ vertexes in the path are $S_{i}^{0}, S_{i}^{1}, ..., S_{i}^{k}$. The total DBW cost for the queries of all pairs of the shortest paths between any two vertexes over the broadcast sequence can be computed as follows:

$$DBW = \sum_{i,j \in V} f(i,j) \ast (\max(\bigvee_{m=0}^{k} \pi(S_{ij}^{m})) - \min(\bigvee_{m=0}^{k} \pi(S_{ij}^{m})))$$

This is essentially the same as the cost for spatial range queries. $f(i,j)$ is equivalent to $w_{ij}, w_{ij,k}, ..., w_{1,2,..,n}$ depending on the number of vertexes along the path between vertex $i$ and $j$. If we group $f(i,j)$ by $k=|S_{ij}|$ and denote this as $f(i,j)^{k}$, then $f(i,j)^{2} \equiv w_{ij}, f(i,j)^{3} \equiv w_{ij,k}, ..., f(i,j)^{n} \equiv w_{1,2,..,n}$.

Similarly, the costs of access time to the data channel under the Multiplexing and Separate Channels schemes for network path query of graph data are as follows correspondingly:

$$AT_{Data}^{Mul} = \sum_{i,j \in V} f(i,j) \ast \max(\bigvee_{m=0}^{k} \pi(S_{ij}^{m}))$$

$$AT_{Data}^{Sep} = \sum_{i,j \in V} f(i,j) \ast g(\max(\bigvee_{m=0}^{k} \pi(S_{ij}^{m})) - \min(\bigvee_{m=0}^{k} \pi(S_{ij}^{m})))$$

Now our problem is how to minimize the DBW and the access time to data under the Multiplexing and Separate channels schemes. To solve this problem, we first present a unified hypergraph representation of the spatial range query for point data and network path query for graph data in Chapter 4. Based on this representation
we relate the optimization problem with the well-known graph layout problems. We then present the optimization methods for the access time under the three cost models which are presented in Chapter 6.

3.4 Discussions on Related Work

The cost model presented in (Chung, 2001) is the work most related to our cost model under the Multiplexing scheme. The cost model is restated as follows using our definitions for the purpose of consistency. Let \( t_j = d_j + \delta_j \) where \( d_j \) is the time to access the \( j^{th} \) required item and \( \delta_j \) is the time between the \( j^{th} \) and \((j+1)^{th}\) required data items in a broadcast channel. Let \( F(y) \) be the access time of query \( q \) begin to access the broadcast channel at time (position) \( y \) in the broadcast sequence. They considered two scenarios when accessing the \( j^{th} \) data item in the broadcast sequence. When a user begins to access the broadcast channel during \( 0 \sim |d_j| \), the user has to wait for the whole broadcast cycle to retrieve \( d_j \), thus \( F(y) = L \). During \( |d_j| \sim t_j \), a user begins to access the channel at position \( y \) during \( \delta_j \) and it takes \( L - y \) units time at most to retrieve the \( j^{th} \) data item, thus \( F(y) = L - d_j \). The average cost for processing the query can be written as follows as derived in (Chung, 2001):

\[
\text{Cost} = \int_{0}^{t_j} F(y) \, dy = \frac{1}{L} \sum_{j=1}^{n} \int_{0}^{t_j} F(y) \, dy
\]

\[
= \frac{1}{L} \sum_{j=1}^{n} \left[ \int_{0}^{|d_j|} F(y) \, dy + \int_{|d_j|}^{t_j} F(y) \, dy \right]
\]

\[
= \frac{1}{L} \left[ \int_{0}^{|d_j|} L \, dy + \int_{|d_j|}^{t_j} (L - y + |d_j|) \, dy \right]
\]

\[
= L - \frac{1}{2L} \sum_{j=1}^{n} (t_j - |d_j|)
\]

\[
= L - \frac{1}{2L} \sum_{j=1}^{n} \delta_j^2
\]
The authors also defined a new measure called Query Distance (QD) to approximate the cost for a complex query \( q \) under ordering \( \pi \) as follows:

\[
QD(q, \pi) = L - \delta_k,
\]

where \( \delta_k \) is the maximum of all \( \delta_j \)s. If \( \delta_k = L - L_2 \) we can see that QD is \( L_2 \). Thus their proposal using QD to approximate the average access time is similar to ours using \( L_2 \) to approximate \( g(L_2) \) as explained in details in Section 6.5.

The authors also claimed that if \( QD(q, \pi_1) \geq QD(q, \pi_2) \), then \( \text{cost}(\pi_1) \geq \text{cost}(\pi_2) \) as the rationale for the approximation. Unfortunately their proof on the induction of “if \( \delta_k(\pi_1) \leq \delta_k(\pi_2) \), then \( \sum_{i=1}^{n} \delta_i(\pi_1)^2 \leq \sum_{i=1}^{n} \delta_i(\pi_2)^2 \) ” is incorrect. A counter example is as follows. Suppose we have only three data items. Their \( \delta_j \)s under \( \pi_1 \) are 2, 4, 4 and 3, 3, 4 under \( \pi_2 \). Thus \( \delta_k(\pi_1) \) and \( \delta_k(\pi_2) \) are both 4. Although we have \( \delta_k(\pi_1) \leq \delta_k(\pi_2) \),

\[
\sum_{i=1}^{n} \delta_i(\pi_1)^2 = 2*2 + 4*4 = 20 \leq \sum_{i=1}^{n} \delta_i(\pi_2)^2 = 3*3 + 3*3 = 18
\]

does not hold in the second induction step.

Since they failed to prove the correctness of using QD \( (L_2) \) to approximate the cost \( (g(L_2)) \), it is natural for readers to question the validity of our cost model and approximation proposal (using \( L_2 \) to approximate \( g(L_2) \)). We next show that under certain circumstances we can reduce their complex cost model to ours and prove the validity of our approximation proposal.

In (Chung, 2001), when \( \delta_k = L - L_2 \) and \( \delta_j = 0 \) (for all \( j=1\sim n \) and \( j \neq k \)), this cost model is essentially the same as ours. Although this condition does not hold in most
cases, when $\delta_k$ is large, $\delta_k^2$ dominates the term $\sum_{j=1}^{n} \delta_j^2$ due to the quadratic relationship and we can use $\delta_k^2$ to approximate $\sum_{j=1}^{n} \delta_j^2$. We believe that the condition can be satisfied in the orderings based on reasonably good heuristics that utilize spatial relationship. For orderings based on these heuristics, data items in the same queries are likely to be close to each other in the orderings which makes $\delta_k=L-L_2$ much larger than other $\delta_i$.

In summary, the work (Chuang 2001) first provided an accurate yet complex cost model. They proposed to use a simpler parameter (QD) to approximate the cost computed by the model. Unfortunately, their proof of the monotonic relationship between QD and the cost model is not correct. On the other hand, we make some approximations (by omitting $L_{2m}$ as discussed in Section 3.1) at the beginning of deriving our cost model. The derived cost model is simpler and easier to compute. In addition, we are able to show the monotonic relationship between $L_2$ and $g(L_2)$ which provides a theoretical foundation for approximation.