Chapter 7

Optimal Query Mapping

Similarly to previous mOLAP systems, *FCLOS* maps incoming queries to the corresponding nodes of the aggregation lattice. Query mapping is employed in order to reduce the number of handled data items and to exploit sub-cube derivability. This chapter deals with the issue of optimal mapping multidimensional queries to aggregation lattices. Previous related work in mOLAP systems has not provided any specific answer to the question of whether the *DCL* or the *hDCL* is optimal for query mapping. We identify the exploitation degree of sub-cube derivability as the critical parameter to be considered. Motivated by this fact, we introduce an analytical framework that facilitates the computation of sub-cube derivability probabilities in aggregation lattices. Besides providing the basis for a general evaluation, the framework focuses on the specific domain of mOLAP [113].

The chapter is structured as follows: Section 7.1 gives the motivation of analyzing optimal query mapping in mOLAP. In Section 7.2, we revise the fundamentals of query mapping. Section 7.3 explains why query mapping is critical in mOLAP systems. In Section 7.4, we introduce an analytical framework for evaluating the exploitation degree of sub-cube derivability in mOLAP systems. Finally, Section 7.5 compares state of the art mOLAP schedulers with respect to query mapping and provides experimental results.

7.1 Motivation

Data broadcast technology plays a fundamental role on wireless dissemination systems, since it is a 1-to-n process, enabling enhanced scalability. Nevertheless, early data broadcast systems were designed under the principle that the number of handled data items is not too high, and that the data items occupy relatively small size and do not possess any semantic connection with each other (e.g., web pages). Content or characteristics of data items were practically ignored. In this context, the dissemination of multidimensional cubes, which are order of magnitude bigger than web pages, and between which semantic connections exist, has to be tackled differently.

To cope with this problem, mOLAP dissemination systems employ query mapping to aggregation lattices in order to reduce the number of handled data items. Moreover, they exploit sub-cube derivability to serve multiple requests with one transmission and thus increase scalability. Since the number of different multidimensional queries that can be issued by the mobile clients is unlimited, the reduction of handled data items is achieved by mapping them to an aggregation lattice, which is a graph representing different views of the data cube. Having mapped queries to lattice nodes, two different queries corresponding to lattice nodes between which a dependency exists, do not have to be served by two separate transmissions, but from a single broadcast instead. All transmissions at the physical layer of the wireless network are anyway broadcast.

There are two types of aggregation lattices, depending on the inclusion of hierarchical levels of dimensions or not. The first discussed type is the *DCL*, which consists of nodes that represent dimensions at their lowest level (fact table data) ignoring hierarchies, leading to *coarse-grained* representations. The second type is the *hDCL*, which includes hierarchies leading to *fine-grained* representations.

Previous work regarding query mapping does not provide a clear answer regarding which aggregation lattice is optimal in mobile DWs. The main reason is that this problem strongly depends on the observed domain. For example, in [69], *hDCL* mapping is used for cache management. In contrast, mOLAP dissemination systems in [141, 139] operate on *DCL* query mapping.

7.2 Derivability in Aggregation Lattices

The characteristics of both aggregation lattices were in detail explained in Section 2.4. Practically, while both lattices provide a cube (and its sub-cubes) representation, the representation of *hDCL* is more detailed. Actually, the *DCL* is a subset of its respective *hDCL*. Figure 7.1 depicts two aggregation lattices. Removing the white nodes of the *hDCL* produces the respective *DCL*.

7.3 Query Mapping in mobile OLAP Architectures

mOLAP architectures serve multidimensional queries issued from mobile clients. They are responsible for the scheduling of queries and the dissemination of results. The entire concept of mOLAP systems is founded on providing offline functionality. This is crucial when considering that portable devices are not permanently connected to a network. Therefore, clients are aware of the schema metadata, can consequently locally store received data and perform local processing in order to enable subsequent analysis.

FCLOS, *STOBS* [139] and *SBS* [141] are mOLAP architectures, explicitly designed for dissemination of multidimensional data in wireless networks. Despite many differences in their scheduling approach, all of them build on *subsumption*. In this way, multiple benefits are gained. The server answers requests faster, and

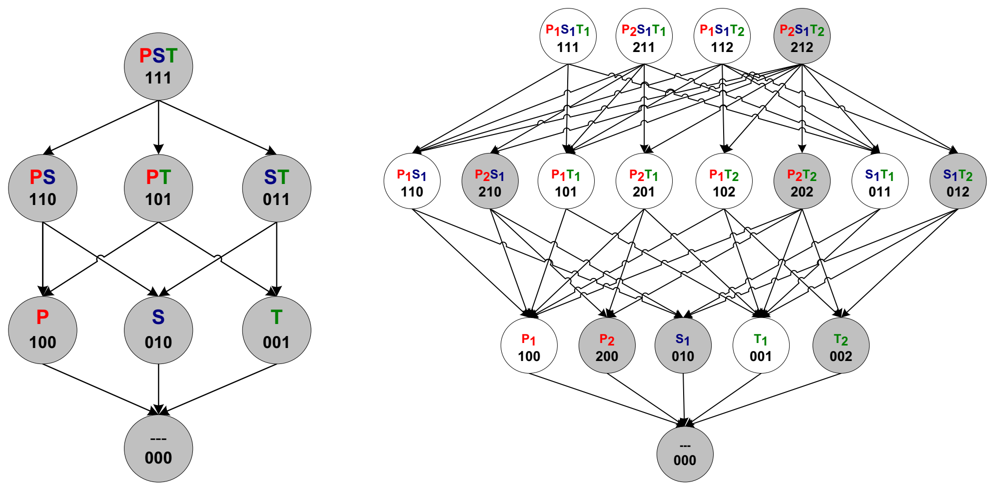

Figure 7.1: Aggregation lattices (DCL on the left, hDCL on the right)

consequently clients experience improved access time as well. Moreover, energy consumption and generated traffic are reduced.

mOLAP systems map incoming queries to the respective nodes of the lattice. This is justified by the fact that the point-to-point communication model is inefficient in wireless networks, especially for OLAP applications. In other words, serving each request individually, assuming that the client is not able to perform local processing, not only exhibits poor performance in terms of access time, but does not scale with the number of requests as well. Query mapping substantially reduces the number of data items that the scheduler has to handle, which not only simplifies, but also assists the operation of broadcast schedulers. Figure 7.2 highlights the question that naturally arises: which of the two lattices should be used for the mapping of the queries? The *DCL* or the *hDCL*? How does this decision influence the system's behavior? An example of query mapping is given in Section 2.6.

Both *FCLOS* and *STOBS* operate on *DCL* mapping. This results in a coarse-grained query mapping. The absence of hierarchies practically imposes that the clients have to locally aggregate fact table tuples in order to compose the hierarchical level aggregations.

Although previous work in this area mainly assumes coarse-grained querying, it is worthwhile investigating the impact of fine-grained query mapping that can be achieved when queries are mapped to *hDCLs*. In fine-grained querying scenarios, transmitted structures include aggregated values according to the dimension hierarchy levels. Such querying imposes that end users receive datasets that require less local processing. Moreover, data transfer in preferred granularities is supported.

Table 7.1 summarizes the tradeoff that arises: On the one hand, using *hDCL* query mapping, generated traffic for a given query is reduced (or is equal in worst

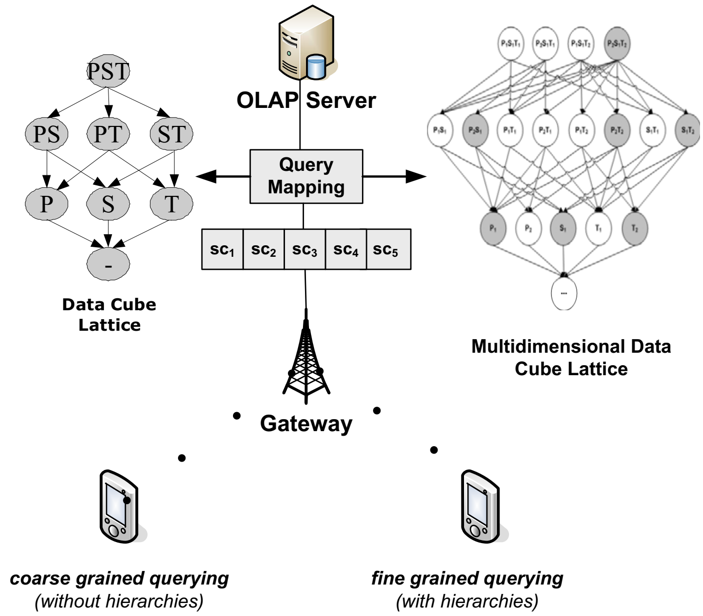

Figure 7.2: mOLAP dissemination system and two possible types of query mapping

Table 7.1: Query mapping tradeoff in mOLAP

Querying/Property	DCL	hDCL
Generated traffic for a given query	-	+
Client side processing	-	+
Number of data items	+	-
OLAP operations	+	-

case). Moreover, clients have to perform less local processing. On the other hand though, sub-cube derivability is more straightforward in *DCL*. The *DCL* consists of fewer nodes than its respective *hDCL*, so that the number of handled data items is reduced. In addition to that, *DCL* mapping enables better offline functionality and OLAP operations like roll-up or drill-down. Consider the query of Section 2.6. A client having received the sub-cube PT can locally (offline) answer any subsequent roll-up, drill-down or projection. A client having received the sub-cube P_1T_1 instead, is not able to locally perform any drill-down. A drill-down would inevitably have to invoke a new query to the server.

7.4 An Analytical Model for Subsumption Probabilities

7.4.1 Model

In the previous section, we considered a mOLAP system and described the trade-offs imposed by selecting between coarse-grained and fine-grained query mapping modes. We suggest that the exploitation of the derivation semantics is the key factor influencing the query mapping mode selection. However, evaluating this exploitation degree is not a straightforward task. This section presents an analytical model that facilitates this evaluation. This model provides the basis for computing relevant subsumption probabilities that mOLAP schedulers have to consider when scheduling and disseminating requests.

mOLAP schedulers, regardless of their individual approach, are keen to exploit subsumptions among the lattice's nodes, but naturally, there are fundamental differences in the way that this is pursued. Therefore, the usefulness of an evaluation probability strongly depends on the scheduler itself. Considering that our objective is to evaluate the suitability of the two different query mapping modes, we do not restrict our model to a specific scheduler. Nonetheless, we show in following paragraphs which of the evaluation probabilities are directly applicable to *STOBS* and to *FCLOS*.

Consider the architecture of Fig. 7.2, where mobile clients issue queries targeting a given data cube. These queries are mapped by the scheduler to the corresponding lattice node and subsequently inserted in the waiting queue. Naturally, one node of the lattice might have been requested by more than one client. Therefore, our model considers the queue as a *multiset* Q and thus $|Q|$ is the size of the queue. Table 7.2 provides a notation overview.

We employ the following probabilities, which can be used by any scheduler, when trying to exploit dependencies:

1. $P(e_a \succeq e_b)$: the probability that a selected element $e_a \in Q$ is an ancestor of another selected element $e_b \in Q$.
2. $P(e_a \succeq Q)$: the probability that a selected element $e_a \in Q$ is an ancestor of every element in Q.
3. $P(\exists\ e : e \succeq Q)$: the probability that there exists one element, which is an ancestor of every element in Q.
4. $P(e_a \succeq q \subseteq Q)$: the probability that a selected element $e_a \in Q$ is an ancestor of exactly $|q|$ $(|q| - 1 + itself)$ elements of Q.
5. $P(e_a \succeq q^+ \subseteq Q)$: the probability that a selected element $e_a \in Q$ is an ancestor of at least $|q|$ $(|q| - 1 + itself)$ elements of Q.
6. $P(\exists\ e : e \succeq q \subseteq Q)$: the probability that there exists at least one element, which is ancestor of exactly $|q|$ $(|q| - 1 + itself)$ elements of Q.

Table 7.2: Notation of the analytical framework

Notation	**Definition**
S	Set that contains every lattice node
Q	Multiset that contains every queue element
$q \in Q$	A subset of Q
D	Number of dimensions
n_j	A lattice node ($n \in S$)
n_j^i	The i_{th} bit of the binary representation of n_j where $0 \leq i \leq D-1$
$n_a \succeq n_b$	n_a is an ancestor of n_b
$n_j \succeq MS$	n_j is an ancestor of every element of the multiset MS (ancestor of $\|MS\|$ elements)
e_j	An element of the queue ($e \in Q$)
gr_d	Number of hierarchical attributes of dimension d
D_n	Set of all dimensions existing in node n
$lev(n,d)$	The hierarchical level of node n at dimension d ($lev(n,d) \in (0, gr_d]$)
P_d	The probability that dimension d exists in node n
$P_{d,l}$	Given that dimension d exists in node n, the probability that the hierarchical level l is selected
p_n	The probability that a node n is selected from a multiset Q of lattice nodes

$P(e_a \succeq e_b)$ for example provides a general evaluation of sub-cube derivability and is not directly related to mOLAP. $P(e_a \succeq q^+ \subseteq Q)$ is intended for any mOLAP scheduler that checks subsumptions after having determined the element to be transmitted, as in the case of *STOBS*. $P(\exists\ e : e \succeq q \subseteq Q)$ on the contrary targets schedulers that consider every element as candidate for transmission and decide according to its subsumption, as in the case of *FCLOS*.

A fundamental objective of this work is to be applicable regardless of the query distribution. Assuming a specific distribution would severely restrict the applicability of our approach. Nevertheless, the discussed probabilities are obviously dependent on the query distribution. We overcome this by allowing the server to collect simple statistics about the incoming queries. Particularly, the server measures the probability P_d that a dimension d exists in an incoming query. For example, using the data cube of Fig. 2.4 the probability $P_{Product}$ that dimension *Product* appears in an incoming query can be computed as $P_{Product} = p_{PST} + p_{PS} + p_{PT} + p_P$.

Additionally, since every dimension has hierarchical levels, for every dimension d, the probability $P_{d,l}$ that the hierarchical attribute l is requested, is also computed by the server. In this way, our model can be applied to any possible query distribution. Naturally, in order to be able to react to fluctuations in the incoming workload, the server will have to keep statistics updated. A simple update formula for the probability $\hat{p}$ is:

k-th observation:

$$\hat{p}_k = \frac{n_k}{n}$$

(k+1)-th observation:

$$\hat{p}_{k+1} = \frac{n_{k+1}}{n+1} = \frac{n_k + x_{new}}{n+1} = \frac{n_k}{n} \cdot \frac{n}{n+1} + \frac{x_{new}}{n+1} = \lambda \hat{p}_k + (1-\lambda) x_{new}$$

where n is the number of events, n_k is the number of occurrences in the k-th observation and x_{new} the possibility that a new event is an occurrence.

In this context, our model considers the probabilities P_d independent. Although this might not be the case for some workloads, it is practically of minor importance for the purpose of the intended comparison. The proposed analytical framework does not directly provide information about expected access time or energy consumption. It delivers a comparative evaluation of the two query mappings.

It should be clear that if we consider every node of the lattice: $\sum_{n \in S} p_n = 1$, and that for every dimension d: $\sum_{l=0}^{gr_d} P_{d,l} = 1$.

7.4.2 Ancestor Probabilities in DCL

A quick way to find out whether a node n_b can be subsumpted by n_a is to apply the binary AND operator on their binary representations (bitmaps):

- If $({n_a}^{bin}\ AND\ {n_b}^{bin}) = {n_b}^{bin}$ then $n_a \succeq n_b$
- If $({n_a}^{bin}\ AND\ {n_b}^{bin}) \neq {n_b}^{bin}$ then $n_a \not\succeq n_b$

In Fig. 2.4 for example, it is $({n_{PS}}^{bin}\ AND\ {n_S}^{bin}) = (110\ AND\ 010) = 010 = {n_S}^{bin}$, which confirms that sub-cube S can be subsumpted by sub-cube PS.

The DCL graph is separated into distinct levels according to the dimensionality of the nodes. We enumerate the levels of the DCL as follows: Nodes with dimensionality l appear in the l_{th} level of the graph. For example the root node with dimensionality D appears at the D_{th} level. Of course $l \in \{0, 1, ..., D\}$. Every level of the DCL consists of $\binom{D}{l}$ nodes. The number of nodes in the DCL is $\sum_{l=0}^{l=D} \binom{D}{l} = 2^D$. A node n_a belonging to the l_{th} level of the graph has $(2^l - 1)$ successors.

Proposition 7.4.2.1. *The probability that a selected element $e_a \in Q$ is an ancestor of another selected element $e_b \in Q$ is :*

$$P(e_a \succeq e_b) = \prod_{d=0}^{D-1} \left(P_d^2 - P_d + 1\right)$$

Proof. We isolate one bit i of the bitmap (which represents one dimension): $P(e_a^i \succeq e_b^i) = P(e_a^i = 0) \cdot P(e_b^i = 0) + P(e_a^i = 1) \cdot P(e_b^i = 0 \vee e_b^i = 1) =$ $(1 - P_i) \cdot (1 - P_i) + P_i \cdot 1 = P_i^2 - P_i + 1$

For the subsumption property to be valid this must be valid for every dimension:

$$P(e_a \succeq e_b) = P((e_a^0 \succeq e_b^0) \wedge \cdots \wedge (e_a^{D-1} \succeq e_b^{D-1})) = \prod_{d=0}^{D-1} \left(P_d^2 - P_d + 1\right)$$

□

Proposition 7.4.2.2. *The probability that a selected element $e_a \in Q$ is an ancestor of every element in Q is:*

$$P(e_a \succeq Q) = \sum_{n \in S} p_n \psi^{|Q|-1} \quad \textit{where } \psi = \prod_{d \notin D_n} (1 - P_d)$$

Proof. For the remaining $(|Q| - 1)$ elements of Q, it suffices that dimensions $d \notin D_{e_a}$ also do not exist in these elements. This is represented by ψ. Thus:

$$P(e_a \succeq Q) = P((e_a \succeq e_0) \wedge \cdots \wedge (e_a \succeq e_{|Q|-1})) = \sum_{n \in S} p_n \psi^{|Q|-1}$$

□

Proposition 7.4.2.3. *The probability that there exists one element, which is an ancestor of every element in Q is:*

$$P(\exists\ e : e \succeq Q) = \sum_{k=1}^{|Q|} \left\{ (-1)^{k+1} \binom{|Q|}{k} \sum_{n \in S} p_n^k \psi^{|Q|-k} \right\}$$

Proof.

$$\begin{aligned} P(\exists\ e : e \succeq Q) &= P(e_0 \succeq Q \vee \cdots \vee e_{|Q|-1} \succeq Q) \\ &= \binom{|Q|}{1} \cdot P(e_a \succeq Q) - \binom{|Q|}{2} \cdot P(e_a \succeq Q \wedge e_b \succeq Q) + \cdots \pm \\ &\quad \binom{|Q|}{k} P(e_0 \succeq Q \wedge \cdots \wedge e_{k-1} \succeq Q) \end{aligned}$$

□

Proposition 7.4.2.4. *The probability that a selected element $e_a \in Q$ is an ancestor of exactly $|q|$ ($|q|-1+itself$) elements of Q is:*

$$P(e_a \succeq q \subseteq Q) = \binom{|Q|-1}{j-1} \cdot \sum_{n \in S} \left(p_n \psi^{j-1} (1-\psi)^{|Q|-j} \right)$$

Proof. $|q|$ elements of Q must be chosen from the 2^l successors and the rest ($|Q|-|q|$) from the ($2^D - 2^l$) non-successors. There are $\binom{|Q|-1}{|q|-1}$ combinations of places in the queue where the ($|q|-1$) successors can be. For ($|q|-1$) elements, we demand that every dimension $d \notin D_n$ does not exist in the examined node. For the rest ($|Q|-|q|$) non-successors, we demand that at least one dimension $d \notin D_n$ does exist in the examined node. □

Proposition 7.4.2.5. *The probability that a selected element $e_a \in Q$ is an ancestor of at least $|q|$ ($|q|-1+itself$) elements of Q is:*

$$P(e_a \succeq q^+ \subseteq Q) = \sum_{j=|q|}^{|Q|} \left\{ \binom{|Q|-1}{j-1} \cdot \sum_{n \in S} \left(p_n \psi^{j-1} (1-\psi)^{|Q|-j} \right) \right\}$$

Proof. $P(e_a \succeq q^+ \subseteq Q) = \sum_{j=|q|}^{|Q|} P(e_a \succeq j \subseteq Q)$ □

Proposition 7.4.2.6. *The probability that there exists at least one element, which is ancestor of exactly $|q|$ ($|q|-1+itself$) elements of Q is:*
$P(\exists\, e : e \succeq q \subseteq Q) = 1 - (1 - P(e_a \succeq q \subseteq Q))^{|Q|}$

Proof. $P(\exists\, e : e \succeq q \subseteq Q) = 1 - P(\overline{\exists\, e : e \succeq q \subseteq Q}) = 1 - P(\overline{e_a \succeq q \subseteq Q})^{|Q|} = 1 - (1 - P(e_a \succeq q \subseteq Q))^{|Q|}$ □

7.4.3 Ancestor Probabilities in hDCL

In this section, we compute the same probabilities as in Section 2.4, but for a multiset Q of *hDCL* elements.

Proposition 7.4.3.1. *The probability that a selected element $e_a \in Q$ is an ancestor of another selected element $e_b \in Q$:*

$$P(e_a \succeq e_b) = \sum_{n \in S} p_n \prod_{d \in D_n} \left(P_d \sum_{l=lev(n,d)}^{gr_d} P_{d,l} + (1 - P_d) \right) \cdot \prod_{d \notin D_n} (1 - P_d)$$

Proof. Analogously to the proof of Proposition 7.4.2.1 we isolate one digit i of the representation. For the subsumption property to be valid for one specific dimension, we demand that in the examined element either this dimensions appear but in a higher, less detailed level or that it does not exist at all.

$$P(e_a^i \succeq e_b^i) = P(e_a^i \geq e_b^i | e_a^i \geq 1) + P(e_b^i = 0 | e_a^i = 0)$$

For the subsumption property to be applied for the element this must be valid for every dimension. We differentiate between dimensions $d \in D_{e_a}$ and dimensions $d \notin D_{e_a}$:

$$P(e_a \succeq e_b) = \prod_{d \in e_a} P_d P((lev(d, e_a) \geq lev(d, e_b > 0)) \vee d \notin D_{e_b}) \cdot \\ \cdot \prod_{d \notin e_a} (1 - P_d) P(d \notin D_{e_b}) = \cdots$$

□

Proposition 7.4.3.2. *The probability that a selected element $e_a \in Q$ is an ancestor of every element in Q is:*

$$P(e_a \succeq Q) = \sum_{n \in S} p_n \omega^{|Q|-1} \quad where$$

$$\omega = \prod_{d \in D_n} \left(P_d \sum_{l=lev(n,d)}^{gr_d} P_{d,l} + (1 - P_d) \right) \prod_{d \notin D_n} (1 - P_d)$$

This can be proven using Proposition 7.4.3.1 for the remaining $(|Q| - 1)$ elements of Q.

Proposition 7.4.3.3. *The probability that there exists one element, which is an ancestor of every element in Q is:*

$$P(\exists\ e : e \succeq Q) = \sum_{k=1}^{|Q|} \left\{ (-1)^{k+1} \binom{|Q|}{k} \sum_{n \in S} p_n^k \omega^{|Q|-k} \right\}$$

The proof is analogous to the proof of Proposition 7.4.2.3.

Proposition 7.4.3.4. *The probability that a selected element $e_a \in Q$ is an ancestor of exactly $|q|$ ($|q| - 1 + itself$) elements of Q is:*

$$P(e_a \succeq q \subseteq Q) = \binom{|Q| - 1}{j - 1} \cdot \sum_{n \in S} p_n \omega^{j-1} (1 - \omega)^{|Q|-j}$$

The proof is analogous to the proof of Proposition 7.4.2.4.

Proposition 7.4.3.5. *The probability that a selected element $e_a \in Q$ is an ancestor of at least $|q|$ ($|q| - 1 + itself$) elements of Q is:*

$$P(e_a \succeq q^+ \subseteq Q) = \sum_{j=|q|}^{|Q|} \left\{ \binom{|Q|-1}{j-1} \cdot \sum_{n \in S} p_n \omega^{j-1} (1-\omega)^{|Q|-j} \right\}$$

Proof. $P(e_a \succeq q^+ \subseteq Q) = \sum_{j=|q|}^{|Q|} P(e_a \succeq j \subseteq Q)$ □

Proposition 7.4.3.6. *The probability that there exists at least one element, which is ancestor of exactly $|q|$ ($|q| - 1 + itself$) elements of Q is:*

$$P(\exists\ e : e \succeq q \subseteq Q) = 1 - (1 - P\left(e_a \succeq q \subseteq Q\right))^{|Q|}$$

The proof is analogous to the proof of Proposition 7.4.2.6.

7.5 Experimental Evaluation

This section provides a twofold analysis. State of the art mOLAP systems are evaluated with respect to query mapping. Moreover, we show how the analytical model described in Section 7.4 facilitates the analysis and confirms the experimental results.

As already explained in Chapter 6, *FCLOS* operates on *DCL* query mapping. However, due to the fact that query mapping is undertaken by the server and not by the clients, different query mappings can be seamlessly integrated without any architectural modification. The scheduling component of *FCLOS* works in exactly the same way with *hDCL* mapping.

The simulation environment is exactly the same used in Section 6.6.1. For the purpose of this chapter, we implemented the extensions of *FCLOS* and *STOBS*, $FCLOS_{hDCL}$ and $STOBS_{hDCL}$, respectively, which map queries to the respective *hDCL*.

7.5.1 Exploitation of Subsumption

Before presenting our experimental results, we explain how our analytical model can be used to facilitate the evaluation. We restrict our analysis to three discussed probabilities. The rest of them provide similar observations.

First, we use Propositions 7.4.2.1, 7.4.3.1 to compute the subsumption probability of two randomly selected elements residing in the scheduler's queue. The probabilities P_d for each dimension are computed from the server's statistics, as described in Section 7.4.1. If no statistics are available, the server has to wait until sufficient statistics about the workload have been collected. The results appear in Table 7.3. *DCL* mapping makes subsumption exploitation almost twice as probable.

Table 7.3: Subsumption probability $P(e_a \succeq e_b)$

	DCL mapping	**hDCL mapping**
Subsumption probability	21.08%	12.77%

This result indicates that *DCL* mapping might be optimal, but our analytical model enables a much more thorough analysis. We consider a waiting queue consisting of $|Q|$ elements. Based on our experimental results we have observed that for a realistic scenario $|Q| \in [20, 40]$. We therefore use $|Q|$=30. We compute the probabilities $P(e_a \succeq q^+ \subseteq Q)$ and $P(\exists\ e : e \succeq q \subseteq Q)$ as defined in Section 7.4.

Figure 7.3 depicts the results of a simple usage of Propositions 7.4.2.5, 7.4.2.6, 7.4.3.5, 7.4.3.6. It is arguably one of the most insightful charts concerning subsumption exploitation by mOLAP systems. This chart has a twofold meaning. On the one hand, as far as query mapping is concerned, we observe the superiority of *DCL* mapping, for both depicted probabilities. When $|q|$ becomes relatively high, as expected, the probability tends to become 0 for both approaches (this can also be derived from Propositions 7.4.2.2, 7.4.3.2). When $|q|$ is smaller though, which represents a more realistic situation, *DCL* mapping outperforms its competitor.

On the other hand, Fig. 7.3 indirectly reveals the superiority of *FCLOS* compared to *STOBS*, this time analytically. This can be observed by isolating the chart series with circular markers representing $P(e_a \succeq q^+ \subseteq Q)$ and the chart series with square markers representing $P(\exists\ e : e \succeq q \subseteq Q)$. As already mentioned, $P(e_a \succeq q^+ \subseteq Q)$ is a useful probability for *STOBS*, which checks derivations only after the element to be transmitted has been selected. In contrast, $P(\exists\ e : e \succeq q \subseteq Q)$ is a useful probability for *FCLOS*. The reason for that is that *FCLOS* considers every element as an ancestor candidate.

Obviously, the superiority of *FCLOS* is even higher, if instead of $P(e_a \succeq q^+ \subseteq Q)$, the probability $P(e_a \succeq q \subseteq Q)$ is computed.

7.5.2 Evaluating mOLAP Dissemination Systems

From an application perspective, query access time is the most important measure for performance evaluation. Figure 7.4 demonstrates that mobile users profit from *DCL* mapping, regardless of the scheduling approach. More specifically, *hDCL* mapping doubles the query access time. Note that the times observed include the necessary time for local processing of data, whenever this is necessary. The results for energy consumption are quite similar.

Figure 7.5 depicts a measure that translates more directly to the information provided by the analytical framework. It shows the percentage of the waiting queries that are served per broadcast, a metric directly related to the results of Fig. 7.3. Particularly for *FCLOS*, the difference is immense. With *DCL* mapping, a broadcast serves half of the waiting queries, whereas with *hDCL* mapping only

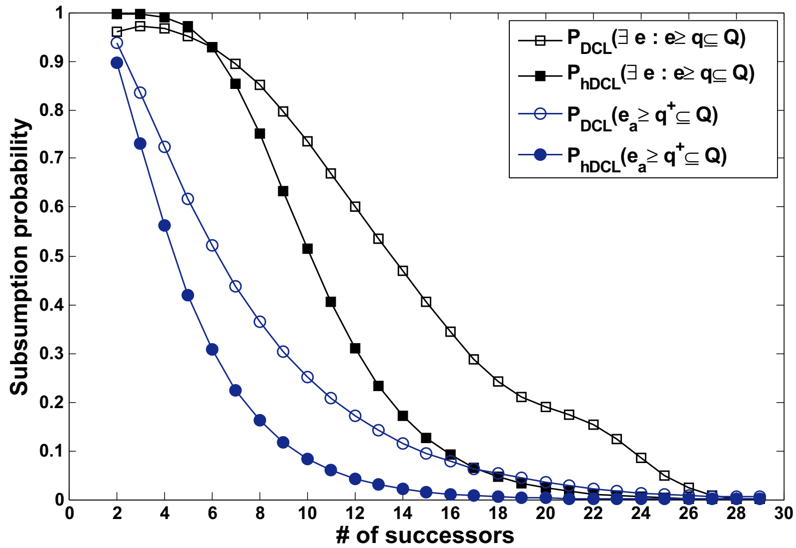

Figure 7.3: Subsumption probabilities $P(e_a \succeq q^+ \subseteq Q)$ and $P(\exists\ e : e \succeq q \subseteq Q)$

25% of them. The difference is significantly smaller in the case of *STOBS*. This is justified by the fact that *STOBS*, as already explained, is not based on subsumption, it just uses subsumption whenever it is possible. In other words, *STOBS* adopts a less aggressive subsumption approach than *FCLOS*, which is confirmed by this chart.

As far as generated traffic is concerned, it is rational to expect more transmitted bytes per broadcast with *DCL* mapping, since *DCL* nodes always contain fact table data that frequently is a superset of what the client had requested. Figure 7.6, depicting the amount of transmitted bytes per broadcast, confirms this expectation. *DCL* mapping provokes transmission of bigger sub-cubes, since *DCL* nodes contain extra information. Practically this means that every transmission lasts longer.

Figure 7.7 on the contrary, depicts the generated traffic per issued query, where a completely different behavior can be observed. *hDCL* mapping appears to cause a huge overhead to the network, especially in the case of *STOBS*. *DCL* mapping does indeed invoke transmissions of more data per broadcast cycle, but manages to serve queries with fewer broadcasts, as indicated by Figure 7.5. With *DCL* mapping there is always a higher probability that the element to be transmitted is an ancestor of other queue elements. One transmission serves more requests, thus reducing the number of necessary broadcast cycles for a given amount of requests.

Concluding the experimental evaluation, it is imperative to underline that the information provided by the proposed framework should be used comparatively. The analytical framework is not capable of predicting the behavior of *FCLOS*, *STOBS* or whichever mOLAP architecture, simply because it was not designed

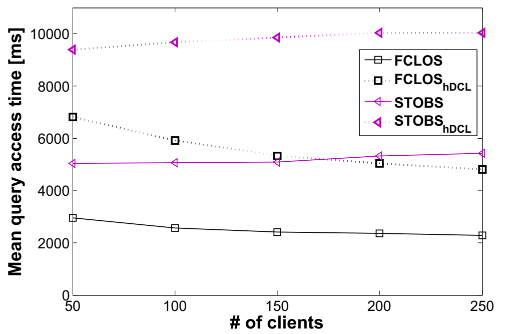

Figure 7.4: Mean query access time (T_{all})

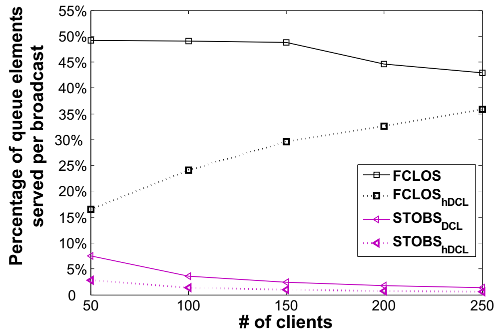

Figure 7.5: Percentage of queue elements served per broadcast

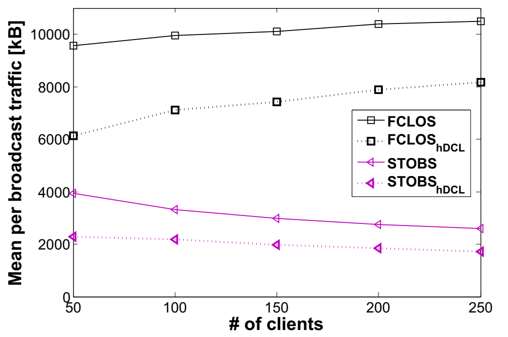

Figure 7.6: Mean per broadcast generated traffic (Tr_b)

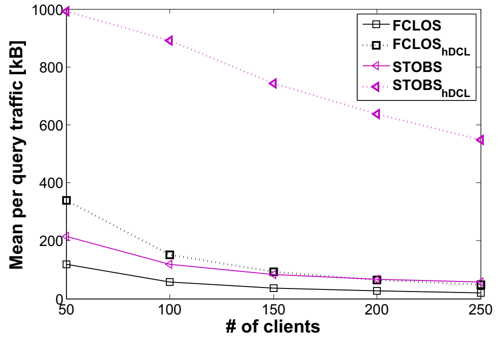

Figure 7.7: Mean per query generated traffic (Tr_q)

for that purpose. Its objective is to provide insight about the exploitation of subsumption probabilities, which is a fundamental, but by no means the only, factor influencing mOLAP performance.

7.6 Summary

This chapter deals with the issue of mapping multidimensional queries to aggregation lattices. mOLAP systems employ query mapping in order to reduce the number of handled data items and to exploit sub-cube derivability. Previous related work in mOLAP systems has not provided any specific answer to the question of whether the *DCL* or the *hDCL* is optimal for query mapping. We identify the exploitation degree of sub-cube derivability as the critical parameter to be considered. Motivated by this fact, we introduce an analytical framework that facilitates the computation of sub-cube derivability probabilities in both lattices. This framework, besides providing the basis for a general evaluation, focuses on the specific domain of mOLAP dissemination systems.

The analytical framework revealed that for a real dataset, sub-cube derivability is optimally exploited in the case of *DCL*. Experimental results for state of the art mOLAP systems confirm that both server and mobile clients benefit from *DCL* mapping. Although with *DCL* mapping, bigger in size datasets are per broadcast transmitted, in comparison with respective *hDCL* mapping, the optimal exploitation of sub-cube derivability results in more clients being served by each transmission, thus reducing the number of necessary broadcasts. The server profits from the reduction of the generated traffic, while mobile clients experience reduced access time and energy consumption.