OLAP over XLM Data

Alfredo Cuzzocrea
ICAR-CNR, Italy & University of Calabria, Italy

INTRODUCTION

Data cubes (Gray et al., 1997) are widely regarded as powerful tools for *OnLine Analytical Processing* (OLAP) (Codd et al., 1993). Based on a *multidimensional* and *multi-resolution* vision of data, data cubes are able of supporting a wide set of analysis methodologies for decision making purposes in a plethora of application scenarios ranging from *Data Warehousing* (DW) to *Decision Support Systems* (DSS) and *Business Intelligence* (BI). The success of OLAP analysis methodologies in the context of *multidimensional data* arising in the above-mentioned application scenarios has given rise to a solid and well-developed technology (Chaudhuri & Dayal, 1997), with important follows in both the academic and industrial research communities.

Symmetrically, *eXtensible Markup Language* (XML) has become very popular in actual *Enterprise Information Systems*, due to its well-understood capabilities and nice amenities. Among these, capturing and modeling typical enterprise data, which are inherently *semi-structured* in nature, plays a leading role. Furthermore, XML is also a dominant *data integration* language/formalism, due to its capability of acting as *lingua franca* for heterogeneous data schemas and formats.

It has been very easy to foresee the marriage between OLAP and XML technologies. In fact, on a side, enterprise data become more and more trendy in the context of next-generation application scenarios where heterogeneity of data and software platforms is the main issue to be faced-off, such as *Web and Grid Service-based Systems*, *P2P Networks*, *Business Process Management Systems*, and so forth. On the other side, OLAP analysis methodologies are able to support critical analysis methodologies over *multidimensional semi-structured data sets* underlying the above-mentioned environments, being these data sets characterized by large volumes, high dimensionality, strong heterogeneity, high correlation. To give some relevant examples, noticeable ones are: *decision making*, *trend analysis*, *time series analysis*, *analytics*, and so forth.

From the convergence of OLAP and XML technologies, a challenging research issue derives: "How to compute an OLAP data cube over XML data?" This problem can be formalized as follows. Given an XML data source X and an OLAP logical schema $\mathcal{L}$ modeling a data cube $\mathcal{D}$, compute from elements in $\mathcal{X}$ the set of SQL-based aggregations populating data cells in $\mathcal{D}$. This implies that a certain SQL aggregate operator $\mathcal{A}$ is given as input. Popular aggregate operators that are widely used in OLAP are:

1. **SUM:** which retrieves the summation of a set of (numerical) XML elements $\mathcal{E}$ stored in $\mathcal{X}$;
2. **COUNT:** which counts the number of elements of a set of (numerical or categorical) XML elements $\mathcal{E}$ stored in $\mathcal{X}$;
3. **AVG:** which retrieves the average value of a set of (numerical) XML elements $\mathcal{E}$ stored in $\mathcal{X}$.

Given an $\mathcal{A}$-based data cube $\mathcal{D}$ over an XML data source $\mathcal{X}$, according to naïve *aggregation schemes* developed in the traditional context of OLAP over relational data, in order to compute the $\mathcal{A}$-based aggregate value of a data cell C in $\mathcal{D}$ (also called *measure*) *all* the XML elements in $\mathcal{X}$ satisfying a certain *multidimensional membership condition W* with respect to a set of at-

DOI: 10.4018/978-1-4666-5202-6.ch150

tributes of analysis (also called *dimensions*) must be accessed and aggregated on the basis of $\mathcal{A}$.

In more detail, let d_i denote an OLAP dimension of an N-dimensional data cube $\mathcal{D}$ over an XML data source $\mathcal{X}$, and $M(d_i) = \{a_{i,0}, a_{i,1}, \ldots, a_{i,Pi-1}\}$ the set of *dimensional members* of d_i, such that $|d_i| = P_i$ denotes the cardinality of d_i. The multidimensional membership condition W corresponds to a JOIN predicate between the N sets of dimensional members, i.e. $M(d_0) \bowtie M(d_1) \bowtie \ldots \bowtie M(d_{N-1})$, which selects the set of XML elements to be aggregated dimensional-member-wise. This finally originates an $\mathcal{A}$-based SQL aggregation that is *univocally* associated to a *multidimensional entry* $\langle a_{0,j}, a_{1,j}, \ldots, a_{N-1,j}\rangle$, such that $0 \leq j \leq P_i \; \forall \; i \in \{0, 1, \ldots, N-1\}$, in the (logical) multidimensional space of $\mathcal{D}$.

While in OLAP over relational data tuples to be aggregated are accessed by means of conventional SQL statements, when OLAP over XML data sources is considered, XML elements must be accessed by means of a standard XML query language, like *XQuery* or *XPath*, via so-called *path-based queries* allowing us to extract the XML elements involved by the OLAP aggregation scheme. This because of the lack of native *grouping constructs* in XML query languages (Paparizos et al., 2002; Beyer et al., 2005; Gokhale et al., 2007). The remaining logical tasks of the relational OLAP aggregation scheme introduced above are, essentially, the same.

The above-described general OLAP aggregation scheme for XML data has been indeed of relevant interest for the Database and Data Warehousing research community since the last decade. The deriving problem, i.e. *computing an OLAP data cube over XML data*, has been regarded as one of the most influent research issue in next-generation DW, DSS and BI applications and systems. This has caused a proliferation of proposals attempting to solve this so-relevant research challenge, and the appearance of the term "*cubing algorithms for XML data*," which identifies algorithms designed to effectively and efficiently compute data cube cells from XML data, given an input OLAP logical scheme (i.e., materializing the target data cube). Under a broader vision, cubing algorithms can also refer other similar activities performed on top of multidimensional data cubes, such as querying data cubes, or processing data cubes. Another related topic is represented by the issue of efficiently querying OLAP data cubes computed over XML data, which should pursue *optimization strategies*, like similar research efforts in "classical" (i.e., relational – e.g., (Cuzzocrea, 2005; Cuzzocrea & Serafino, 2009)) or "advanced" (i.e., stream-based – e.g., (Cuzzocrea et al, 2004; Cuzzocrea et al., 2005)) OLAP data cubes.

Inspired by this fundamental motivation, this chapter proposes a survey on cubing algorithms for XML data, along with a critical analysis of these algorithms, and discussion on open issues and future research trends in XML-OLAP research.

A SURVEY ON CUBING ALGORITHMS OVER XML DATA SOURCES

As highlighted in Section 1, the problem of effectively and efficiently computing a data cube over an XML data source has been of great interest through the Database and Data Warehousing research community. Here, a comprehensive and critical survey on state-of-the-art proposals appeared in literature so far is proposed.

(Jensen et al., 2001a; Jensen et al., 2001b) first discusses the problem of how to specify OLAP data cubes over XML data. To this end, (Jensen et al., 2001a; Jensen et al., 2001b) proposes a UML-based methodology for specifying cubes and other typical OLAP constructs like hierarchies associated to data cube dimensions and OLAP aggregations. In addition to this, a framework intended to integrate XML and relational data in order to support typical OLAP operations and activities over Web-distributed data is presented.

(Niemi et al., 2002) moves the attention to the problem of constructing OLAP data cubes from