IRM PRESS
701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.irm-press.com
ITB9198

Chapter V

Building Signature-Trees on Path Signatures in Document Databases

Yangjun Chen*
University of Winnipeg, Canada

Gerald Huck
IPSI Institute, Germany

ABSTRACT

Java is a prevailing implementation platform for XML-based systems. Several high-quality in-memory implementations for the standardized XML-DOM API are available. However, persistency support has not been addressed. In this chapter, we discuss this problem and introduce PDOM (persistent DOM) to accommodate documents as permanent object sets. In addition, we propose a new indexing technique: path signatures to speed up the evaluation of path-oriented queries against document object sets, which is further enhanced by combining the technique of signature-trees with it to expedite scanning of signatures stored in a physical file.

INTRODUCTION

With the rapid advance of the Internet, management of structured documents such as XML documents has become more and more important (Suciu & Vossen, 2000; World Wide Web, 1998a; Marchiori, 1998). As a subset of SGML, XML is recommended by the W3C (World Wide Web Consortium) as a document description metalanguage to

** The author is supported by NSERC 239074-01 (242523) (Natural Science and Engineering Council of Canada)*

exchange and manipulate data and documents on the WWW. It has been used to code various types of data in a wide range of application domains, including a Chemical Markup Language for exchanging data about molecules and the Open Financial Exchange for swapping financial data between banks, and between banks and customers (Bosak, 1997). Also, a growing number of legacy systems are adapted to output data in the form of XML documents.

In this chapter, we introduce a storage method for documents called *PDOM* (persistent DOM), implemented as a lightweight, transparent persistency memory layer, which does not require the burdensome design of a fixed schema. In addition, we propose a new indexing technique: *path signatures* to speed up the evaluation of path-oriented queries against document object sets, which are organized into a tree structure called *a signature-tree*. In this way, the scanning of a signature file is reduced to a binary tree search, which can be performed efficiently. To show the advantage of our method, the time complexity of searching a signature-tree is analyzed and the permanent storage of signature-trees is discussed in great detail.

BACKGROUND

The Document Object Model (DOM) is a platform- and language-neutral interface for XML. It provides a standard set of objects for representing XML data: a standard model of how these objects can be combined and a standard interface for accessing and manipulating them (Pixley 2000). There are half a dozen DOM implementations available for Java from several vendors such as IBM, Sun Microsystems and Oracle, but all these implementations are designed to work in main memory only. In recent years, efforts have been made to find an effective way to generate XML structures that are able to describe XML semantics in underlying relational databases (Chen & Huck, 2001; Florescu & Kossmann, 1999; Shanmugasundaram et al., 1999; Shanmugasundaram & Shekita, 2000; Yosjikawa et al., 2001). However, due to the substantial difference between the nested element structures of XML and the flat relational data, much redundancy is introduced, i.e., the XML data is either flattened into tuples containing many redundant elements, or has many disconnected elements. Therefore, it is significant to explore a way to accommodate XML documents, which is different from the relational theory. In addition, a variety of XML query languages have been proposed to provide a clue to manipulate XML documents (Abiteboul et al., 1996; Chamberlin et al., 2001; Christophides et al., 2000; Deutsch et al., 1989; Robie et al., 1998; Robie, Chamberlin & Florescu, 2000). Although the languages differ according to expressiveness, underlying formalism and data model, they share a common feature: *path-oriented queries*. Thus, finding efficient methods to do path matching is very important to evaluation of queries against huge volumes of XML documents.

SYSTEM ARCHITECTURE

The system architecture can be pictorially depicted as shown in Figure 1, which consists of three layers: persistent object manager, standard DOM API and specific PDOM API, and application support.