Chapter V

XML Stream Query Processing: Current Technologies and Open Challenges

Mingzhu Wei
Worcester Polytechnic Institute, USA

Ming Li
Worcester Polytechnic Institute, USA

Elke A. Rundensteiner
Worcester Polytechnic Institute, USA

Murali Mani
Worcester Polytechnic Institute, USA

Hong Su
Oracle Cooperation, USA

ABSTRACT

Stream applications bring the challenge of efficiently processing queries on sequentially accessible XML data streams. In this chapter, the authors study the current techniques and open challenges of XML stream processing. Firstly, they examine the input data semantics in XML streams and introduce the state-of-the-art of XML stream processing. Secondly, they compare and contrast the automaton-based and algebra-based techniques used in XML stream query execution. Thirdly, they study different optimization strategies that have been investigated for XML stream processing – in particular, they discuss cost-based optimization as well as schema-based optimization strategies. Lastly but not least, the authors list several key open challenges in XML stream processing.

Copyright © 2009, IGI Global, distributing in print or electronic forms without written permission of IGI Global is prohibited.
INTRODUCTION

In our increasingly fast-paced digital world all activities of humans and surrounding environments are being tagged and thus digitally accessible in real time. This opens up the novel opportunity to develop a variety of applications that monitor and make use of such data streams, typically stock, traffic and network activities (Babcock et al., 2002). Many projects, both in industry and academia, have recently sprung up to tackle newly emerging challenges related to stream processing. On the academic side, projects include Aurora (Abadi et al., 2003), Borealis (Abadi et al., 2005), STREAM (Babu & Widom, 2001), Niagara (Chen et al., 2002), TelegraphCQ (Chandrasekaran et al., 2003), and CAPE (Rundensteiner et al., 2004). On the industrial side, existing major players in database industry such as Oracle (Witkowski et al., 2007) and IBM (Amini et al., 2006) have embarked on stream projects and new startup companies have also emerged (Streambase, 2008; Coral8, 2008).

While most of these activities initially focused on simple relational data, it is apparent that XML is an established format and has been widely accepted as the standard data representation for exchanging information on the internet. Due to the proliferation of XML data in web services (Carey et al., 2002), there is also a surge in XML stream applications (Koch et al., 2004; Florescu et al., 2003; Diao & Franklin, 2003; Bose et al., 2003; Russell et al., 2003; Ludascher et al., 2002; Peng & Chawathe, 2003). For instance, a message broker routes the XML messages to interested parties (Gupta & Suciu, 2003). In addition, message brokers can also perform message restructuring or backups. For example, in an online order handling system (Carey et al., 2002), suppliers can register their available products with the broker. The broker will then match each incoming purchase order with the subscription and forward it to the corresponding suppliers, possibly in a restructured format at the request of the suppliers. Other typical applications include XML packet routing (Snoeren & Conkey, 2001), selective dissemination of information (Altinel & Franklin, 2000), and notification systems (Nguyen et al., 2001).

XML streams are often handled as a sequence of primitive tokens, such as a start tag, an end tag or a PCDATA item. To perform query evaluation over such on-the-fly XML token streams, most systems (Diao et al., 2003; Gupta & Suciu, 2003; Ludascher et al., 2002; Peng & Chawathe, 2003) propose to use automata to retrieve patterns from XML token streams. However, although automata is a suitable technique for matching expressions, how to improve and extend automata functionality in order to efficiently answer queries over XML streams has been a topic of active debate by the XML community. Further, one distinguishing feature of pattern retrieval on XML streams is that it relies solely on the token-by-token sequential traversal. It is not possible to jump to a certain portion of the stream (analogous to sequential access on magnetic tapes). Thus, the traditional index-based technologies cannot be applied for effective query optimization. In static XML processing, cost-based and schema-based optimization techniques are widely used. How to perform such optimization and other optimization techniques in the streaming XML context is a major challenge, and is thus one of the topics of this chapter.

Roadmap. XML streams can be viewed in different data granularities (i.e., different data models for XML streams), which is discussed in Section 2. In Section 3 we describe the automaton-based and algebra-based techniques used in XML stream query processing. Cost-based and schema-based optimization techniques are presented in Section 4 respectively. Section 5 lists the open challenges in XML stream processing and concludes this work.