Chapter 4
Foundation of XML Stream Processing Techniques

Weidong Yang
Fudan University, China

Hao Zhu
Fudan University, China

ABSTRACT
The problem of processing streaming XML data is gaining widespread attention from the research community, and various XML stream processing methods are put forward, including automaton-based methods, index-based methods, and so forth. In this chapter, the basic concepts and several existing typical approaches of XML stream processing are discussed. Section 1 introduces the background and current research status of this area. Section 2 focuses on the discussion of automaton-based methods, for example, X/YFilter, XPush, et cetera. In section 3, the index-based methods are given. In section 4, other methods such as Fist and XTrie are discussed briefly. Section 4 discusses some optimization technique of XML stream processing. Section 5 summarizes this chapter.

4.1 INTRODUCTION

4.1.1 Background

Nowadays, XML is considered as a standard of data representation and exchange, it is widely used in various fields such as SOAP and WSDL web services, business to business transactions and personalized content delivery. On the other hand, due to the demands from sensor network applications, information dissemination, content-based routing, and processing of scientific data, efficient processing of XML stream data has attracted lots of attentions in recent years.

A data stream is defined as a massive unbounded sequence of data elements continuously generated at a rapid rate. A data stream management system (DSMS) registers queries in advance because it should execute them continuously perpetually rather than once on demand. The registered query is called a continuous query. It produces its query result whenever an incoming tuple of a target data stream satisfies it. Accord-
ingly, it should be evaluated in real-time, which requires strict time and space constraints.

Particularly, a data stream which is formatted by XML is called an XML Stream, its corresponding management system is called XML stream management system. If we treat XML as a tree, XML stream is a string sequence after a preorder traversal in this tree. XML stream data can be transmitted by two different ways: one is by character stream, the commoner one is by token or chunk. A token is made by several elements, and transmit together.

XML data is consisting of many nesting elements, which each one begins with a certain open tag and ends with the corresponding close tag. XML data is ordered, hierarchical and maybe recursive, so its processing is more complicated than the ordinary data. A well-formed XML stream is one in which every pure XML stream which is delimited by control tokens is a well-formed XML fragment. Most researches only consider well-formed streams and fragments. Well-formed XML fragments correspond to labeled ordered trees, with labels of inner nodes indicating the element name of a node and leaf nodes representing character content.

Generally, there are two different method to process XML data, DOM (Document Object Model) and SAX (Simple API for XML). DOM treats XML data as an entire element tree, and keeps the whole tree in the memory for processing. Unlike DOM, SAX is an event-driven API for processing XML. Obviously, SAX is more suitable for processing XML stream data than DOM. In the most researches, XML stream data is modeled as a stream of SAX events: startElement(id, tag, text) and endElement(id, tag), where tag is the tag of the node being processed, text is text value of the node in the corresponding XML tree, and id is unique identifier of the node. By implementing these interfaces, whenever the open tag or close tag is reached, the above two events will be activated to processing the data.

For XML stream management systems, currently, most queries are expressed by some structured query languages such as XPath (XQuery 1.0, 2011) or (Su, Jinhui, & A. Rundensteiner, 2003). The core technical challenge in such systems is processing a large set of XPath queries over a continuously incoming XML stream. This challenge is related to, but different from, the more traditional stored XML data retrieval problem. XML stream is generated at a rapid rate and arrives continuously, therefore it requires the processing system has a better processing capability and lower memory used. XML streaming data are available for reading only once and are provided in a fixed order determined by the data source.

4.1.2 The State of Art

Several relational data stream management systems have been proposed since 2002, such as Aurora (Abadi et al., 2003), Telegraph (Madden, Shah, & Hellerstein, 2002) and STREAM (Motwani, et al., 2003). Similar to these systems, XML stream filtering is also based on the concept of continuous queries processing for streaming data. But XML data is ordered, recursive and hierarchical, mostly queried by XPath/XQuery language. The differences between XML data and ordinary relational data determine that the special techniques are needed for processing XML stream data.

Several studies have been concentrated on how to evaluate multiple and large amount of XPath queries over XML stream data. University of California at Berkeley proposed XFilter (Altinel & Franklin, 2000) at 2000 in the VLDB (one of the most famous conference in the field of database), which builds a finite state machine (FSM) for each XPath query and employs a query index on all the FSMs. As the successor of XFilter, YFilter (Diao, & J. Franklin, 2003) combines all of the XPath queries into a single NFA (non-deterministic finite automaton), supporting shared processing of the common prefixes among all navigation...
Related Content

An Efficient Approach for Incremental Association Rule Mining through Histogram Matching Technique
www.irma-international.org/article/efficient-approach-incremental-association-rule/74782/

Ontology-Driven Keyword Search for Heterogeneous XML Data Sources
www.irma-international.org/chapter/ontology-driven-keyword-search-heterogeneous/69126/

Decomposed PRNU Library for Forensics on Photos
Yue Li (2013). Modern Library Technologies for Data Storage, Retrieval, and Use (pp. 129-142).
www.irma-international.org/chapter/decomposed-prnu-library-forensics-photos/73770/

Efficient Retrieval Technique for Microarray Gene Expression
www.irma-international.org/article/efficient-retrieval-technique-microarray-gene/74783/

The Use of Mixed Methods in Organizational Communication Research
www.irma-international.org/chapter/use-mixed-methods-organizational-communication/69732/