Fast Answering of XPath Query Workloads on Web Collections

Mariano P. Consens and Flavio Rizzolo

University of Toronto
{consens,flavio}@cs.toronto.edu

Abstract. Several web applications (such as processing RSS feeds or web service messages) rely on XPath-based data manipulation tools. Web developers need to use XPath queries effectively on increasingly larger web collections containing hundreds of thousands of XML documents. Even when tasks only need to deal with a single document at a time, developers benefit from understanding the behaviour of XPath expressions across multiple documents (e.g., what will a query return when run over the thousands of hourly feeds collected during the last few months?). Dealing with the (highly variable) structure of such web collections poses additional challenges.

This paper introduces DescribeX, a powerful framework that is capable of *describing* arbitrarily complex XML summaries of web collections, enabling the efficient evaluation of XPath workloads (supporting all the axes and language constructs in XPath). Experiments validate that DescribeX enables existing document-at-a-time XPath tools to scale up to multi-gigabyte XML collections.

1 Introduction

Web applications rely heavily on XML tools to manipulate data encoded in XML. Data can be exchanged, as in web feeds (blogs, news feeds, podcasts) or via web service messages. Data can also be stored, as in hypertext collections like Wikipedia. Several XML manipulation tasks (and the tools used to implement them) process one document at a time, whether the document is an individual RSS file, a single SOAP message, or a Wikipedia article in XHTML. The vast majority of software tools utilized in this context rely on XPath as the core dialect for XML querying. Hence, web developers make extensive use of embedded XPath queries for processing XML collections.

A developer working with this type of collection faces several challenges. She must learn enough about the (semi)structure present in the XML collection to be able to write meaningful XPath queries. She must also develop an understanding of how the XPath expressions behave across different documents in the collection.

Understanding the actual structure of a web collection can be a significant barrier. Some collections (like Wikipedia or personal blogs) do not really have a schema, or the schema allows most elements to occur almost anywhere. Even when XML documents are validated against a proper schema, their actual structure can vary significantly across the collection. This can happen because the

D. Barbosa et al. (Eds.): XSym 2007, LNCS 4704, pp. 31–45, 2007.

schema is large and only small (possibly disjoint) subsets are actually used (as happens with industry standard schemas, like IXRetail[1]), or because schemas can be arbitrarily composed using open content models (e.g. RSS extensions like Yahoo Media, podcasts, etc.). In these scenarios, schemas alone are not that helpful for understanding (nor for optimizing) XPath evaluation.

This paper argues that DescribeX, a tool supporting powerful structural summaries, can help with understanding the (semi)structure of large collections of XML documents. In fact, DescribeX summaries contribute to significantly speed up (and scale up) XPath evaluation with existing file at a time tools, enabling fast exploration of the results of XPath workloads on large collections.

XML structural summaries are graphs representing relationships between sets in a partition of XML elements. DescribeX summaries have a unique capability: they are the first ones to *describe* precisely the structural commonality that determine each individual set in the partition. DescribeX introduces a language of *axis path regular expressions* (*AxPREs*, for short) to describe the sets.

Most of the existing summary proposals define all sets in the partition using the same criteria, hence creating *homogeneous* summaries. These summaries are based on common element paths (in some cases limited to length k), whether incoming paths [7,11,17], both incoming and outgoing paths [12,21], or sequence of outgoing paths (common subtrees) [3]. The few examples of *heterogeneous* (adaptive) summaries [5,23] have no capability for describing the partitions, which are defined according to very simple criteria (e.g., just the incoming paths).

In contrast, DescribeX supports constructing heterogenous summaries where each set in the partition can be created according to *explicit criteria* obtained from an expression in the complete XPath language (all the axes, document order, use of parenthesis, etc.). Given an arbitrary XPath query, DescribeX can create a partition defined by an AxPRE that captures exactly the structural commonality expressed by the query.

This paper presents experimental results that demonstrate that using a summary created from a given workload can produce query evaluation times orders of magnitude better than using existing summaries. The experiments also validate that DescribeX summaries allow file at a time XPath processors to be a competitive alternative (in terms of performance) to DB-like XML query engines – even on gigabyte sized collections.

Overview and Contributions. The next section walks through a concrete example to illustrate how DescribeX summaries can help developers understand the behaviour of XPath queries across large XML collections. The following two sections present the main technical contributions of the paper. Section 3 provides an overview of the rich framework for describing summaries underlying the DescribeX tool (based on the novel technique of applying bisimilarity to element neighborhoods described by an AxPRE). Section 4 gives a translation from XPath expressions into AxPREs, hence supporting the creation of summaries with nodes that answer complex XPath expressions. The system contributions

[1] `http://www.nrf-arts.org/`