Available online at www.sciencedirect.com
ScienceDirect
Information Systems 33 (2008) 155–181
www.elsevier.com/locate/infosys

An analysis of XQuery benchmarks

Loredana Afanasiev*,[1], Maarten Marx

University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

Abstract

This paper presents a survey and an analysis of the XQuery benchmark publicly available in 2006—XMach-1, XMark, X007, the Michigan benchmark, and XBench—from different perspectives. We address three simple questions about these benchmarks: How are they used? What do they measure? What can one learn from using them? One focus of our analysis is to determine whether the benchmarks can be used for micro-benchmarking. Our conclusions are based on an usage analysis, on an in-depth analysis of the benchmark queries, and on experiments run on four XQuery engines: Galax, SaxonB, Qizx/Open, and MonetDB/XQuery.
© 2007 Published by Elsevier B.V.

Keywords: XQuery; Benchmarks; Micro-benchmarks

1. Introduction

In this paper, we provide a survey and an analysis of the XQuery benchmarks publicly available in 2006: XMach-1 [1], XMark [2], X007 [3], the Michigan benchmark (MBench) [4], and XBench [5].[2] We believe that this analysis and survey are valuable for both the developers of new XQuery benchmarks and for (prospective) users of the existing benchmarks. Henceforth, we call the above mentioned five benchmarks simply *the benchmarks*. We now introduce the questions we address in this study.

How are the benchmarks used? We first look at the actual use of the benchmarks in the scientific community, as reported in the 2004 and 2005 proceedings of the ICDE, SIGMOD and VLDB conferences. Fewer than $\frac{1}{3}$ of the papers on XML query processing that provide experimental results use the benchmarks. The remaining papers use ad hoc experiments to evaluate their research results. Section 2 contains the results of this literature survey.

One of the reasons for this limited use might be the current state of the benchmarks: 29% of the queries in the benchmarks contain errors or use outdated XQuery dialects. We have corrected these errors and rewritten all queries into standard W3C XQuery syntax and made these updated queries publicly available. Section 3 describes the kind of errors we encountered and the way we corrected them.

Another reason that the benchmarks are not widely used might be that the benchmarks do not

*Corresponding author.
E-mail addresses: lafanasi@science.uva.nl (L. Afanasiev), marx@science.uva.nl (M. Marx).
[1]Supported by the Netherlands Organization for Scientific Research (NWO) under Project no. 612.000.207.
[2]X007 is technically not an XQuery benchmark, since its queries are expressed in a predecessor of XQuery.

0306-4379/$ - see front matter © 2007 Published by Elsevier B.V.
doi:10.1016/j.is.2007.05.002

provide suitable measures for the intended purpose of the surveyed experiments. Most of the experiments apply *micro-benchmarking* to test their research results. Micro-benchmarks, as opposed to *application benchmarks*, are benchmarks that focus on thoroughly testing a particular aspect of the query evaluation process, such as the performance of a query optimization technique on a particular language feature [6]. Out of the five benchmarks only the Michigan benchmark (MBench) was designed for micro-benchmarking [4]. The rest of the benchmarks are application benchmarks. We will investigate in detail the micro-benchmarking properties of the MBench queries in Section 5.

What do the benchmarks measure? Our goal is to discover the rationale behind the collection of queries making up a benchmark. Having all queries in the same syntax makes it possible to analyze them systematically. We address this subject with three follow-up questions: (1) How representative of the XQuery language are the benchmark queries?, (2) How concise and focused is the set of queries making up a benchmark?, (3) Are the MBench queries adequate for micro-benchmarking? The first two questions are answered in Section 4; the third, in Section 5.

To answer question (1), we look at functional coverage. The goal is to see how much of XQuery functionality is used in comparison to XPath functionality. If we consider only the retrieval capabilities of XQuery, only 16 of the 163 benchmark queries could not be expressed in XPath 2.0. This indicates that the benchmark queries are heavily biased toward testing XPath rather than XQuery.

For question (2), we analyze each benchmark as a whole. Running and interpreting each query in a benchmark can be costly. So we analyzed the additional value of each query given the rest of the collection. Roughly, a query has no additional value if there is another query that yields the same execution times on all documents on all XQuery processing engines. Our preliminary results indicate that for most of the benchmarks, a well-chosen subset of its queries give the same information as the entire benchmark.

We give a partial answer to question (3) by looking at the MBench queries that test the performance of value-based joins. Based on initial experiments, we identify two different sub-tasks with the evaluation of joins expressed in XQuery: one is to implement an efficient algorithm for value-based joins, and the other is to detect when a query contains a value-based join. XQuery is syntactically a rich language, and joins can be expressed in many different ways. Thus, the second task is not a trivial one. As a follow up work of our investigation and a slight deviation from the main goal of this paper, we extend the set of join queries of MBench to a micro-benchmark testing for the join detection of an engine. We present this micro-benchmark in detail and apply it on four different engines.

What can we learn from using these benchmarks? We found that benchmarks are especially suitable for finding the *limits* of an engine. Though scientific papers rarely give error analyses, an analysis of the errors raised by an engine—syntax, and out-of-memory, and out-of-time—is very useful. Even on our rewritten queries, all the engines except SaxonB raised syntax errors, which indicates their non-compliance to the W3C XQuery standard. We found that the most informative manner of experimenting with and learning from benchmarks is by running them on several engines and *comparing* the results. Such comparisons are meaningful only if they are executed under the same conditions for each engine; for example, without modifying the syntax of the query when running them on different engines. Section 6 contains an analysis of the errors we found in the benchmark queries and a sample of comparison results.

All the experiments in this paper are run with the help of a testing platform, XCheck[3] [7], on four XQuery engines: Galax [8], SaxonB [9], Qizx/Open [10], and MonetDB/XQuery [11].

To summarize, our main contributions are the following:

- Survey of the use of XQuery benchmarks in the scientific literature (Section 2).
- Standardization of the current XQuery benchmark queries (Section 3).
- Analysis of the benchmark queries: What do they measure? (Section 4)
- A micro-benchmark (extending the set of join queries of MBench) testing join detection (Section 5).
- An example of a comparative benchmark study on four XQuery engines (Section 6).

Our conclusions are in Section 7.

[3] http://ilps.science.uva.nl/Resources/XCheck/

دریافت فوری ← متن کامل مقاله

ISIArticles
مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات