A study of gossip algorithms for internet-scale cardinality estimation of distributed XML data

Slavov, Vasil Georgiev

View/Open

- SlavovStuGosAlg.pdf (680.6Kb)

Date
2012-10-03

Format
Thesis

Abstract
After more than a decade of active research and development, the peer-to-peer (P2P) computing model continues to be successful. We have witnessed the deployment of commercial P2P applications in large, Internet-scale environments. With the rise and growth of P2P, indexing and querying data stored in large-scale sharing systems has become increasingly difficult. Computing statistics over data stored in Internet-scale P2P systems is an important component of query optimization. Decentralized gossip-based protocols are very popular in networking, and in particular, in sensor networks. The simplicity and scalability of gossip protocols render them perfect for quickly computing accurate estimates of aggregates (sums, averages, etc.) in Internet-scale systems where node and link failures are the norm. In this thesis, we present the problem of cardinality estimation of XPath queries over XML data stored in a distributed, Internet-scale environment. We focus our work on three objectives: implementing gossip in an Internet-scale environment, conducting a comprehensive performance evaluation in a wide-area network, and analyzing the experimental results. We implement two gossip-based algorithms (VanillaXGossip and XGossip) which, given an XPath query, estimate the number of XML documents in the network that contain a match for the query. XGossip employs a new, divide-and-conquer strategy for load-balancing and reducing the bandwidth consumption. We conduct a comprehensive performance evaluation of both gossip algorithms on Amazon Elastic Compute Cloud (Amazon EC2) web service using a heterogeneous collection of XML documents. The goal of the performance evaluation is to find if the results we obtain are consistent with the theoretical analysis of VanillaXGossip and XGossip.

Table of Contents
Introduction -- Background and motivations -- The design of VanillaXGossip and XGossip -- Implementation of VanillaXGossip and XGossip -- Evaluation -- Conclusion and future work -- Appendix A. Algorithms -- Appendix B. XPath grammar

URI
http://hdl.handle.net/10355/15636

Degree
M.S.

Thesis Department
Computer Science (UMKC)

Collections
Computer Science and Electrical Engineering Electronic Theses and Dissertations (UMKC)
2012 UMKC Theses - Freely Available Online
efficiently compute quantities that take the form of an average over pairs of observations, also known as U-statistics [12]. We consider the problem of estimating set-expression cardinality in a distributed streaming environment where rapid update streams originating at remote sites are continually transmitted to a central processing system. At the core of our algorithmic solutions for answering set-expression cardinality queries are two novel techniques for lowering data communication costs without sacrificing answer precision. Our first technique exploits global knowledge of the distribution of certain frequently occurring stream elements to significantly reduce the transmission of element state information to the central processing system. Results of our experimental study with real-life as well as synthetic data sets indicate that our distributed Abhinandan Das, Sumit Ganguly, Minos N. Garofalaki. Results: Here, we present ntCard, a streaming algorithm for estimating the frequencies of k-mers in genomics datasets. At its core, ntCard uses the ntHash algorithm to efficiently compute hash values for streamed sequences. We have compared the performance of ntCard and other cardinality estimation algorithms. We used three datasets of 480 GB, 500 GB and 2.4 TB in size, where the first two representing whole genome shotgun sequencing experiments on the human genome and the last one on the white spruce genome. Thus, our benchmarks demonstrate ntCard as a potentially enabling technology for large-scale genomics applications. Availability And Implementation: ntCard is written in C++ and is released under the GPL license.