A Practical Evaluation of Load Shedding in Data Stream Management Systems for Network Monitoring

Jarle Søberg, Kjetil H. Hernes, Matti Siekkinen, Vera Goebel, and Thomas Plagemann

University of Oslo, Department of Informatics
P.O. Box 1080, Blindern, N-0316 Oslo
{jarleso, kjetih, siekkine, goebel, plageman}@ifi.uio.no

Abstract. In network monitoring, an important issue is the number of tuples the data stream management system (DSMS) can handle for different network loads. In order to gracefully handle overload situations, some DSMSs are equipped with a tuple dropping functionality, also known as *load shedding*. These DSMSs register and relate the number of received and dropped tuples, i.e., the *relative throughput*, and perform different kinds of calculations on them. Over the past few years, several solutions and methods have been suggested to efficiently perform load shedding. The simplest approach is to keep a count of all the dropped tuples, and to report this to the end user. In our experiments, we study two DSMSs, i.e., TelegraphCQ with support for load shedding, and STREAM without this support. We use three particular network monitoring tasks to evaluate the two DSMS with respect to their ability of load shedding and performance. We demonstrate that it is important to investigate the correctness of load shedding by showing that the reported number of dropped tuples is not always correct.

Keywords: DSMS applications, data stream summaries, data stream sampling

1 Introduction

In this paper, we focus on network monitoring as the application domain for *data stream management systems* (DSMSs). More specifically, we evaluate the load shedding mechanisms of existing DSMSs in the case of *passive network monitoring* where traffic passing by an observation point is analyzed. This field of research is commonly challenged by the potentially overwhelming amounts of traffic data to be analyzed. Thus, data reduction techniques are important. Many of those techniques used by the traffic analysis research community as stand-alone tools are also utilized extensively for load shedding within the DSMSs. Examples include *sampling* [8, 9], *wavelet analysis* [11], and *histogram analysis* [23]. Monitoring tasks may also include timeliness constraints: Storing measurements and performing "off-line" analysis later is impossible in many monitoring scenarios where analysis needs to happen in near real-time. For example, an intrusion detection system (IDS) based on passive traffic measurements needs to generate alerts immediately in the case of a detected intrusion. Internet service providers (ISPs) are interested in real-time measurements of their backbones for traffic engineering purposes, e.g., to re-route some traffic in case congestion builds up in certain parts of the network. In addition, network traffic rarely generates a constant stable input stream. Instead, traffic can be bursty over a wide range of time scales [14], which implies that the monitoring system should be able to adapt to a changing load. For all these reasons, it is very interesting to study the applicability of DSMSs equipped with load shedding functionalities in this context.

The STREAM [1, 3, 4] and Gigascope [7, 6] projects have made performance evaluations on their DSMSs for network monitoring. In both projects, several useful tasks have been suggested for network monitoring. Plagemann et al. [17] have evaluated an early version of the TelegraphCQ [5] DSMS as a network monitoring tool by modeling and running queries and making a simple performance analysis.

With respect to load shedding, Golab et al. [10] sum up several of the different solutions suggested in the DSMS literature. Reiss et al. [22] evaluate sampling, multi-dimensional histograms,