Statistical Science
2006, Vol. 21, No. 4, 476–478
DOI: 10.1214/088342306000000619
Main article DOI: 10.1214/088342306000000583

Comment: Monitoring Networked Applications With Incremental Quantile Estimation

Lorraine Denby, James M. Landwehr and Jean Meloche

Monitoring networked applications is indeed a challenge, particularly so for large networks. There are many types of specific networked applications, and a range of statistical and computational issues comes up in different problems. Chambers et al. discuss the constraints and their solution for a specific business application software problem they faced. While we would like to know more about the real-world nature of their application and how the users' needs and the application's technology drive the constraints they faced, the statistical approach they developed and its results are impressive. Problems that we have faced have somewhat different technological and statistical needs, however. We would like to take this opportunity to describe some general issues and approaches that we believe are useful for networked application monitoring problems.

Given the quality of today's networks and applications with relatively little downtime and stable behavior, the monitoring process often amounts to filtering through large amounts of irrelevant data and looking for the unusual. The monitoring system can be regarded as a compression engine that takes the large amounts of irrelevant data and distills them into the few elements of information that do matter.

The raw data that come into the compression engine originate at multiple agents in the network, often but not exclusively at the endpoint devices of the network where the application clients run. Additional data can also be available from the network elements (e.g., routers and links) that actually handle the traffic. The ultimate destination is often but not necessarily a centralized system where it will be possible to launch corrective action as need be. The frequency with which the raw data arise, the number of agents, the speed at which unusual events are reported, and the size of the reported information are all examples of technical parameters of the monitoring system that are at the root of the challenge.

In every instance of this problem, the technical parameters are constrained by business requirements. The business requirements aim to specify characteristics of the monitoring system such as the speed with which it will notice developing problems, the frequency of false alarms, and the network overhead (amount of network traffic dedicated to the monitoring itself) of the system. The network overhead is an especially difficult constraint to handle in general because it relates to specific characteristics of the network on which the distributed application is running and it requires cost considerations. For example, a large overhead on an optical segment of the network may be of no consequence in comparison to a moderate one on a low-bandwidth link.

For some combinations of the technical parameters and business constraints, it may be possible to simply send all of the raw data to a central server for analysis. The paper that we are discussing starts from the premise that this is not the case. The problem then faced is to design something like a distributed compression machine. Chambers et al. propose a model in which the agents perform part of the compression and send partly summarized data to a central server where the aggregation of the various summaries takes place. First, the agents fill a data buffer D of size N; second, when D is full a quantile buffer Q is updated and D is flushed; third, periodically or upon request, Q is sent to the server for aggregation.

The problem is related to that of distributed source coding that has been studied in computer science. In that context, the agents transmit coded signals to a server where the decoding takes place. The primary goal is to find coding methods that result in a tolerable distortion.

As a first specific and important example, we consider VoIP (Voice over IP). VoIP typically involves a pair of IP phones that send an RTP (real-time protocol) stream to each other. The RTP stream is a sequence of numbered UDP (user datagram protocol)

Lorraine Denby, James M. Landwehr and Jean Meloche are with the Data Analysis Research Department, Avaya Labs, Basking Ridge, New Jersey 07920, USA (e-mail: ld, jml, jmeloche@research.avayalabs.com).