Tribeca: A Stream Database Manager For Network Traffic Analysis

Mark Sullivan

Bell Communications Research
Morristown, NJ 07960
sullivan@bellcore.com

High speed computer and telephone networks carry large amounts of data and signalling traffic. The engineers who build and maintain these networks use a combination of hardware and software tools to monitor the stream of network traffic. Some of these tools operate directly on the live network; others record data on magnetic tape for later offline analysis by software. Most analysis tasks require tens to hundreds of gigabytes of data. Traffic analysis applications include protocol performance analysis, conformance testing, error monitoring and fraud detection.

Although these programs query large databases of network traffic, traffic engineers have usually written their own ad-hoc analysis programs. They avoid conventional DBMS software because of performance concerns and a semantic mismatch between the analysis operations and the operations supported by commercial DBMSs. In traffic analysis, both the data and the storage medium are stream-oriented. Fast sequential access to data is crucial; such features as transactional updates, fast access to random records, and concurrency control are not. The format of data in the stream is defined by network protocol hierarchies which do not easily map to a relational schema. Also, network traffic traces contain many small records with fields a few bits wide, so per-tuple or per-field overheads can noticeably increase the database size.

The traffic analysts use operators more like those found in sequence and temporal DBMSs than relational ones. Analysts search for simple events, aggregate events (such as unusual bursts of packets), or calculate different kinds of network utilization over successive time periods or time scales. They usually run batches of related queries during a single pass over the data. Users sometimes write queries that use partial results generated by a concurrently executing query. Even unrelated queries often share subqueries because of the single data source.

Tribeca is a software system for querying arbitrarily long streams of information from a live network feed, from tape, or from disk. Tribeca reads a stream of data from a single source and applies compiled queries to the stream. Bellcore's traffic analysts prefer it to their ad-hoc programs because, like a relational DBMS, it has a high level query language. Unlike conventional systems, Tribeca does not support expensive features such as random access to data or transactional updates that are unnecessary for this workload. Tribeca is also extensible; it has a type system oriented towards network protocols and user-defined operators so it can integrate specialized traffic analysis operators.

Tribeca's query language is data-flow-oriented. It allows users to construct large batch queries for the single pass over the data. The language supports sequence operations such as window aggregates and operations for demultiplexing and multiplexing substreams derived from the source. It also has a limited form of join that compares an operand small enough to fit in memory to a stream derived from the source. The query language is designed to prevent users from expressing queries that produce intermediate results that cannot be stored in main memory. Query optimization focuses on memory management and predicate ordering. Bellcore traffic analysts have been using a prototype version of Tribeca to analyze recordings of ATM, SS7, and frame relay networks. This is joint work with Andrew Heybey.

Proceedings of the 22nd VLDB Conference
Mumbai(Bombay), India, 1996