High density compression of log files

Balázs RÁCZ[1,2]   and   András LUKÁCS[2]
bracz@math.bme.hu   alukacs@sztaki.hu

[1] Institute of Mathematics, Budapest University of Technology and Economics
Egry József u. 1. H. ép., 1111 Budapest, Hungary
[2] Informatics Laboratory, Computer and Automation Research Institute of the Hungarian
Academy of Sciences, Kende u. 13-17., 1111 Budapest, Hungary

Today there is an emerging demand of Internet and network related Service and Content Providers to collect the valuable service usage data and process it using data mining methods to answer questions regarding security, service improvement or financial issues. Keeping these log data sets in a storage-efficient and easily accessible way suitable for direct processing by data mining algorithms is a challenging problem.

The aim of storing the log data of large volume implies the need of a good time-space tradeoff for encoding placing slightly more emphasis on compression rate. Taking into consideration the requirements of data mining applications we found that decompression should be fast, and it should support a rough random-access of the records.

We present a generalized scheme for preprocessing and high density compression of log files. The aim of the method is to provide a base for long-term storage in a form appropriate for direct processing by data mining algorithms.

Experimental runs on real log data show that our *differentiated semantic log compression* (dslc) methods compress at 2–3%, outperforming general purpose compression utilities, e.g. bzip2, in some cases by a factor of 10. The encoding of dslc is also near 3 times faster than with bzip2, and the time is dominated by the preprocessing phase. Decompression of dslc is as fast as gzip.

The efficiency in compression rate is achieved by *field-wise* compression starting with *semantic compression* and followed by well-chosen compression algorithms suiting the corresponting field. The complex combination of semantic techniques and common general compression algorithms perform better for log compression than standard general purpose compression methods.

The framework presented is extremely modularized and configurable, so processing different log sources can be done with reused components, thus minimalizing the effort needed to integrate new logs into the system, and opening the possibility for plugging in enhanced or specialized compression modules for certain fields. The different semantic and standard compression components are organized into modular *pipelines*. A pipeline for a specific field is composed of standard encoding schemes preceded by *invertible transformations* customizable for the actual data set's actual field. We demonstrate the flexibility of the pipeline concept by inlaying a novel compression algorithm to improve the compression efficiency we reached using well-known methods. We found that the use of the pipeline concept in the implementation leads to a flexible, widely applicable, robust and scalable system for the largest set of log data. Our implementation was designed for the largest Hungarian Internet content provider and fulfills the additional requirements of production environments.