ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

An Efficient Storage and a Method Of Network Monitoring Data Compression

Sivasakthi[1], Juveriya[2]

M.E, Department of CSE, P.S.V College of Engineering and Technology, Krishnagiri, Tamil Nadu, India[1]

Asst. Prof, Department of CSE, P.S.V College of Engineering and Technology, Krishnagiri, Tamil Nadu, India[2]

Abstract: With the increasing sophistication of attacks, there is a need for network security monitoring systems that store and examine very large amounts of historical network flow data. An efficient storage infrastructure should provide both high insertion rates and fast data access. Traditional row-oriented Relational Database Management Systems (RDBMS) provide satisfactory query performance for network flow data collected only over a period of several hours. In many cases, such as the detection of sophisticated coordinated attacks, it is crucial to query days, weeks or even months' worth of disk resident historical data rapidly.For such monitoring and forensics queries, row oriented databases become I/O bound due to long disk access times. To overcome these problems we propose a new column oriented storage infrastructure for network flow records, called NetStore. NetStore is aware of network data semantics and access patterns, and benefits from the simple column oriented layout without the need to meet general purpose RDBMS requirements. The prototype implementation of NetStore can potentially achieve more than ten times query speedup and ninety times less storage size compared to traditional row-stores, while it performs better than existing open source column-stores for network flow data.

Keywords: Network Monitoring, Network Measurements, Traffic Analysis, Space Utilization, Monitoring Data Compression

I. INTRODUCTION

Telecom Operators have to constantly monitor the network for a number of task like management, provisioning, service offering, etc.. Such as billing require the analysis of the data in real time. Network provisioning is performed on a set of statically indicatorsindicators calculated over the last month and year. These task monitoring infrastructures have been presented such as relying on data reduction techniques to keep the amount of data manageable. Similarly these issues have been early addressed by the network research community.

The technology evaluation has required more aggressive lossy approaches such as adaptive shedding of input data. The availability of network monitoring data has allowed for more complex analyses. An analysis of the scalability to the log size. These tasks often require saving the monitoring data. It implies storing for a long time a more amount of information. Telecom operator willing to analyse service access patterns could easily be confronted with large data sets to be transmitted from monitoring points to processing sites and several months.

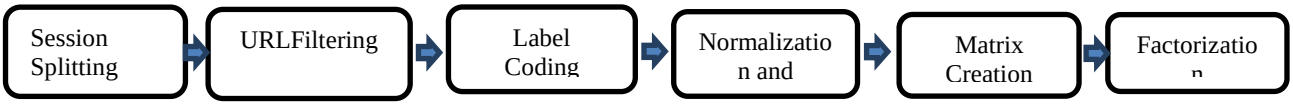

Fig. 1 Architecture Design

SESSION SPLITTING

When dealing with a continuous stream of monitoring data, there is the necessity to split it in parts (chunks) with a finite number of sessions in order to apply the subsequent mapping and processing. The possibility to have overlapping between subsequent chunks or forcing all of them to be disjoint. Allowing for fixed- or varying-size chunks according to different stopping criteria. The data visible to the approximation algorithm, and thus both the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

compression efficiency and the approximation error. The scope of the operations that can be performed directly on the representation. The splitting can be performed before or after the URL filtering. , a fixed split size parameter is used to control the number of sessions allowed for each run of the approximation algorithm.

URL FILTERING

The proposed technique provides the possibility to control the presence of HTTP sessions with rare servers. It means of a URL filtering threshold parameter, defined as the minimum number of occurrences a URL must exhibit in the log in order to be considered. It also affects some sources.

LABEL CODING

The field URL (destination URLs) is stored as a list of unique elements. URLs are ranked by number of occurrences. It mapped to an integer equal to their rank order. a shorter code is assigned to more frequent items. The values in the field source IP are stored as a list with unique elements. The label corresponding to a source IP is given by its position. The load field is already numerical in nature. This way the size of mapping affects the total size of the representation within a percentage that decreases with the growth of the number of sessions.

NORMALIZATION AND WEIGHTING

This is done by dividing the code (ranking) by the total number of srcIDs. The specific application of the technique, different degrees of approximation can be tolerated on the different fields. A valuable property for this optimization is that, differently from the session split process.

TIMESTAMP

The sequence number of the sessions. We can easily calculate the maximum error we can have with this approximation. A = t/N where N is the number of sessions in the observed time window.

Thus, A = T/N where N is the number of sessions in the observed time window and T is the width of the time window, that is affected by the *split size* parameter. The timestamp of entry k is replaced by:

$$ik = ti + (k - 1) * A$$

MATRIX FORMAT

The last step of the preprocessing phase we build P, which is an M x N matrix, representing N sessions (with N equal to the split size parameter), each of which with M dimensions, depending on the number of fields. It could consider other flow parameters as well). In this representation, each column of the input matrix is a vector (srcID; URLcode; Bytes)T where the time is implicitly represented by column index, namely the ith column holds values of the ith entry of the input file.

$$pk = \sum_{i}^{K} ck(i)ti$$

$$\|Pi - Y, ci(l)ti - ci(k)tk\|l + A\|ci(k)\|0$$

Defining e_{ikk} as the residual $p_i - J2_l=_k c_i(l)t_l$, the minimization becomes:

$$\min_{Ci(k)} \|eik - ci(k)tk\|2 + A\|ci(k)\|0$$

Thus reducing the storage requirement and also the computational complexity required by operations on the compressed representation.

FACTORIZATION

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

To store P efficiently, we want to use a matrix P such that: P approximates P,and P = TC,where T(M x K) and C(K x N) are two matrices with high sparsely and thus requiring a small amount of memory for their storage. K is a parameter that is optimized when T and C are derived. K can be larger than M if this decreases the number of non-zero components in C. As our objective is to trade off computational complexity vs. accuracy, we model our approximation as the minimization problem:

P — TC\\\ + \(\\T ||o + \\C\\o)

It takes into account the number of non-zero elements of T and C, it yields sparse results, thus reducing the storage requirement and also the computational complexity required by operations on the compressed representation.

ALGORITHM AND TECHNIQUES

FACTORIZATION ALGORITHM

These requiring a small amount of memory for their storage. The optimization problem takes into account the number of non-zero elements the storage requirement and also the computational complexity required by operations on the compressed representation. The other phases are negligible with respect to the ones required for the factorization. It is necessary to have the complete log file before starting.

COMRESSION ALGORITHM

The total size of the data in the new representation format (we also call it compressed version), as well as the size of specific components, is compared against the size of the original data. CCS: size in bytes of the Compressed Column Sparse representation of the C matrix alone. Tot: sum of the size in bytes of: CCS, T matrix, bzip2-compressed ordered list of URLs, and bzip2-compressed ordered list of source IDs. These components, with a few metadata (namely the four integers representing the dimensions of the matrices and one float per field representing its scaling factor), are all we needed to rebuild the original data (URL-filtered and approximated).

MODELS AND DEFINITIONS

PREPROCESSING

This perform computations directly on the representation, we apply a technique which converts the log in numerical matrices, where each row/column is a vector of real numbers. Our technique comprises two main phases. The first one is called pre-processing phase, and provides as output a fully numeric matrix representation of the original data. The second phase is called factorization phase and it transforms the numeric matrix into a couple of sparse matrices which approximate our original data, while having a smaller memory footprint.

LABEL CODING

The values in the field URL (destination URLs) are stored as a list of unique elements; URLs are ranked by number of occurrences, and mapped to an integer equal to their rank order: the higher the frequency, the higher the

numerical label assigned to them. In a typical encoding, a shorter code is assigned to more frequent items. However, in our case, small values might be changed to zero when attempting to find a sparse representation and we would like to preserve the values of the most frequently accessed items.

SENSITIVITY AND SCALABILITY ANALYSIS

To analyse the sensitivity of the technique to the weighting factors. More specifically, the purpose of introducing the weighting factors in the technique (i.e. allowing for a trade-off between accuracy on a specified field and compression efficiency or accuracy on other fields) is validated. We performed tests with several different weighting.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

PERFORMANCE EVOLUTION

To evaluate the performance of our technique, we have considered its efficiency in terms of memory footprint reduction (Compression Ratio) and different consequences of the approximation (Bytes error, ID error, URL error). For the evaluation of the memory footprint, the compression ratio is compared with the output of the general purpose compression utility bzip2, employing the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding.

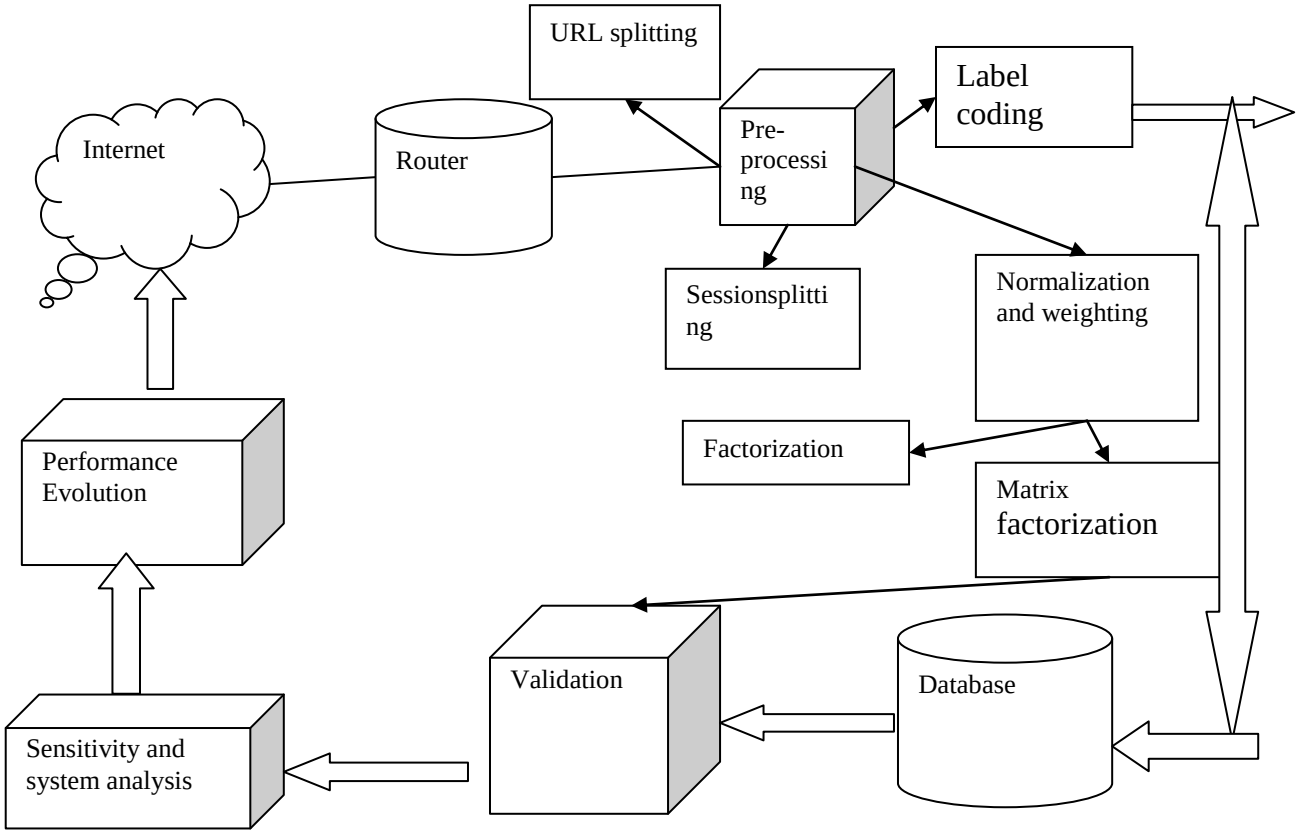

Fig 2. Running Process

II. EXPERIMENTAL EVALUATION

In terms of memory footprint reduction *(*Compression Ratio*)* and different consequences of the approximation *(*Bytes error, ID error, URL error).

TABLE I: Characteristics of traffic traces

	LAB	MAWI
sessions	26965	105310
duration (s)	3599	899
URLs	1771	5926
IDs	110	12129

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

The evaluation of the memory footprint, the compression ratio is compared with the output of the general purpose compression utility *bzip2*, employing the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding.

The reported results are obtained by varying the value of *A*,that controls the amount of distortion allowed in the approximated factorization algorithm (the higher the value of *A*, the higher the tolerated approximation, but also the more sparsity induced in the representation matrix): *A* is varied in the set {0.001,0.0025,0.005, 0.01,0.025,0.05,0.1, 0.25,0.5,*1}*.The analysis are conducted with a fixed URL threshold.

III. Results And Discussion

The technique is suitable for efficient space usage, as can be see in Fig 3, where for fine-grained approximation the resulting space occupation becomes 21% of the space occupation of the original LAB trace and 27% of the original MAWI trace, and compression level equivalent to the bzip2 is reached.

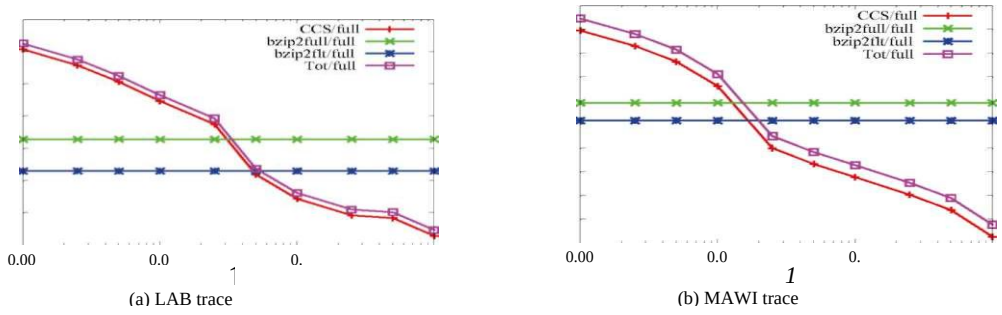

Fig 3. Compression Ratio

This result confirmed for two significantly different traces in terms of length and heterogeneity of IDs and URLs, is of notable value compared to bzip2. Fig 4.

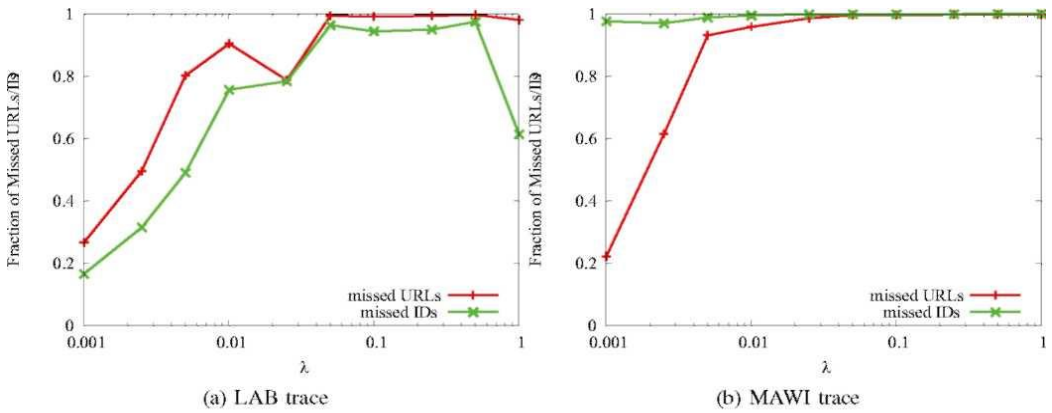

Fig. 3: Fraction of Incorrect URLs and IDs.

Fig 4: Fraction of Incorrect URLs and IDs

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

IV. CONCLUSION

The growth of the telecommunication networks and user base, the need for efficiently collecting, transferring, processing and storing network monitoring data.A technique that stores high-volume network monitoring data in a representation format that satisfies two main objectives at the same time. The efficient utilization of storage space. The possibility to perform a number of operations in a computationally convenient way and directly on the transformed data. The properties are traded-off with a controlled loss of accuracy.

The project explained in detail the different phases composing out technique and the various input parameters, which allow fine-tuning the achievable performance according to different kinds of applications. It also conducted an in-deep evaluation of the technique using data from two different operational networks.

REFERENCES

1. Aceto.G, Botta, A. Pescap´e.A and C. Westphal, "An efficient storage technique for network monitoring data," in 2011 IEEE InternationalWorkshop on Measurements & Networking.
2. Agrawal, S Kanthi, C. N. Naidu K. V. Ramamirtham.M., J. Rastogi,RSatkin, S and A. Srinivasan, "Monitoring infrastructure for convergednetworks and services," Bell Labs Technical J., vol. 12, no. 2, pp. 63–77,2007.
3. Arsa K. K. and Sanguanpong.S "In-memory URL compression," in Proc. 2001 National Computer Science and Engineering Conference, pp. 425–428.
4. Cho.CMitsuya.K and Kato.A, "Traffic data repository at the WIDE project," in Proc. 2000 USENIX Annual Technical Conference.
5. Peuhkuri. M, "A method to compress and anonymize packet traces," in Proc. 2001 ACM Internet Measurement Workshop, pp. 257–261.
6. Ros, P. B. Iannaccone .G.,.Cuxart, J. S D. A. L´opez, and Pareta, J. S. "Load shedding in network monitoring applications," in Proc. 2007 USENIX Annual Technical Conference.
7. Shieh J. and Keogh. E. J. "iSAX: indexing and mining terabyte sized time series," in KDD, Y. Li, B. Liu, and Sarawagi, S editors. ACM, 2008, pp. 623 631.
8. Skibi´nski. PandSwacha.J"Fast and efficient log file compression," in Proc. CEUR Workshop, 2007 East-European Conference on Advances in Databases and Information Systems.