Communication Network Discovery through Correlation Matrix

Guan Ti\textsuperscript{1*}, Geng Yujie\textsuperscript{1}, Ma Qiang\textsuperscript{1}, Liu Yong\textsuperscript{1} and Lin Lin\textsuperscript{1}

\textsuperscript{1} State Grid Shandong Electric Power Company, Jinan, Shandong, 250001, China

*Corresponding author’s e-mail: sddl_zdh@163.com

Abstract. Communication network discovery is a herculean task for IDC operators, who usually been lacking of the knowledge of business architecture. The traditional method, by analyzing the source address and the destination address inside of the communication packet, to explore the network connection graph is extremely inefficient, especially when dealing with the hybrid network. This paper presented a novel approach to discover the communication network through the correlation matrix of the cpu utilization dataset.

1. Introduction

The servers deployed in the same IDC are connected to each other via network devices and communication cables. Considering the business architecture designed in the online system, the graph of the server communication is required to ensure the quality of the services efficiently. For example, the node $A$ and node $B$ in the IDC are connected through the network devices, but may be without exchange of business packet. In this situation, the relationship between node $A$ and node $B$ is helpless for the IDC operator to monitor the server load and the network load. To ensure the quality of the business services, the operators have to identify the relationships of servers, which truly exchange data package between each other. This paper presented a novel approach to find out the connections of the servers, without the access privilege of the network devices, but only with the collected cpu utilization metrics.

2. Related Work

The work related to this paper could be categorized into two groups, the network analysis and the temporal analysis.

2.1. Network Analysis

One research area of network analysis is large network structures exploratory. A great work of the network analysis is the Stanford Network Analysis Platform (SNAP) \cite{1}, a general-purpose, high-performance system that provides easy-to-use, high-level operations for analysis and manipulation of large networks. Another research area is focusing on the analysis of multilayer networks, which are an important way to represent a large variety of complex systems throughout science and engineering. A certain number of works have been proposed on this area. One of which is the “muxViz” \cite{2}, demonstrated to analyse and interactively visualize multilayer data using empirical genetic, neuronal and transportation networks.
2.2. Temporal Analysis
The volume of data being collected, analysed, and stored has exploded in recent years, in particular in relation to the activity on the monitoring system. Several methodologies for constructing models for optimizing the performance of big datasets monitoring have been proposed. Such as Ameen Alkasem et al. [3] proposed a model of real-time monitoring for big dataset analytics using apache spark. Of these monitoring metrics, the cpu usage is the most commonly used metric for performance monitoring. Stefanov K S and Gradskov A A [4] proposed a method to analysis the cpu usage and their impact on performance monitoring.

3. Proposed Method
The paper proposed a method to discover the communication network in the IDC. The method applies the correlation matrix algorithm on the collected cpu utilization monitoring dataset, while no need for the knowledge of the business architecture and the access privilege of the network devices. The pipeline of the method is described in the following.

3.1. Data Collection and Processing
The cpu usage data is the only one metric needed to perform the method. There are many ways to collect the cpu utilization data, however, all of the methods use temporal format to organize the cpu usage data. A temporal data denotes the evolution of an object characteristic over a period of time. The cpu utilization data is a sort of time dependent data, characterized by data elements being a function of time, and defined in form of \( d = f(t) \). For data defined at discrete time stamps \( t_i \), this relation can be represented as: \( D(j) = \{ (t_1, d_1), (t_2, d_2), ..., (t_n, d_n) \} \), where \( d_i = f(t_i) \) and \( D(j) \) is the \( j \)th sample, with \( 1 \leq j \leq m \), \( m \) is the number of samples collected.

For the collected cpu usage samples \( D = \{ D(1), D(2), ..., D(m) \} \), each of the samples \( D(j) \) may have dependent time stamp, \( t_i \), which is \( D(j)(t_i) \neq D(k)(t_i) \), where \( 1 \leq j \leq m \), \( 1 \leq k \leq m \), \( j \neq k \). Before feeding the samples dataset into the correlation matrix algorithm, the dataset should be processed by the normalization method. The main process of the method is mapping the original time stamps \( t_i \) of all the samples \( D \) into one normalized time stamps \( t_i' \), which is: \( d_i' = f(t_i') = \frac{1}{n} \sum f(t_i) \). In other words, the metric \( d_i' \) correlated to the time stamp \( t_i' \) is the average of the metrics \( d_i \) with correlated time stamps \( t_i' \leq t_i < t_i'+1' \). After the normalization process, the original sample \( D \) is converted into \( D' = \{ D'(1), D'(2), ..., D'(m) \} \).

3.2. Network Discovery
The goal of our method is constructing the communication structure from temporal monitoring datasets. This paper leveraged the correlation matrix algorithm on the temporal dataset to explore the network structure of the monitored server nodes. A correlation matrix is a table showing correlation coefficients between sets of variables. Each random variable \( (X_i) \) in the table is correlated with each of the other values in the table \( (X_j) \). This allows you to see which pairs have the highest correlation. The formula used in this method is defined as:

\[
\rho_{X,Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E(X)^2}\sqrt{E(Y^2) - E(Y)^2}}
\]

3.3. Network Visualization
Through the correlation matrix of the monitored server nodes, a network structure can be explored. For complex network, a visualization technique is required to cognize the structure. Network visualizations usually depict nodes as dots or disks and links as arcs connecting the corresponding nodes; conversely, this visualization style is known as node-link diagram. Furthermore, nodes are usually labelled with the name of the entity that they represent. This is necessary to relate nodes in the visualization to nodes in the actual network. Visual variables like color or size can be used to encode additional node properties.
4. Case Study
To evaluated the method proposed in this paper, a case study is organized based on real world situations. The original cpu usage monitoring dataset is collected through Prometheus, a monitoring system with a multi-dimensional data model with time series data identified by metric name and key/value pairs.

The dataset used in the case study is collected from 8 server nodes, during a past period of 4031 time stamps, and only the cpu usage metric is retrieved from the Prometheus databases. The original dataset monitoring visualization is shown in Figure 1. As shown in Figure 1, the 8 server nodes are labeled as ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’ and ‘H’, the correlated values of each node are encoded by connected lines and rendered with a unique color.

Figure 1: CPU Utilization Metric

Then the dataset is normalized into a standard dataset, which is suit to apply the correlation matrix algorithm. The correlation matrix of the 8 server nodes is shown as Table 1 and Figure 2. The each value in the correlation matrix present the correlation coefficient between each pair of nodes, where the value 1 means positive correlation, to the value 0 means no correlation at all. Filtered by the threshold 0.4, the network connection can be visualized in Figure 3. As revealed in Figure 3, two server nodes groups are separated by layout and color, and node “B” has strong connection to node “C” than node “A”.

Table 1: Correlation Matrix of the 8 Server Nodes

|     | A           | B           | C           | D           | E           | F           | G           | H           |
|-----|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| A   | 1.00000000  | 0.64763793  | 0.60478303  | -0.01402055 | 0.20158812  | -0.1678115  | 0.2184517   | 0.14199932  |
| B   | 0.64763793  | 1.00000000  | 0.99146902  | -0.10780875 | 0.11553151  | -0.0523829  | 0.1827728   | 0.18521773  |
| C   | 0.60478303  | 0.99146902  | 1.00000000  | -0.10845100 | 0.12354475  | -0.04489523 | 0.1837586   | 0.18472201  |
| D   | -0.01402055 | -0.10780875 | -0.10845100 | 1.00000000  | -0.06654165 | 0.07831839  | 0.1232602   | -0.05693928 |
| E   | 0.20158812  | 0.11553151  | 0.12354475  | -0.06654165 | 1.00000000  | 0.41675349  | 0.6062065   | 0.33280826  |
| F   | -0.1678115  | -0.0523829  | -0.04489523 | 0.07831839  | 0.41675349  | 1.00000000  | 0.3628664   | 0.19407739  |
| G   | 0.2184517   | 0.1827728   | 0.18375856  | 0.12326021  | 0.60620654  | 0.36286635  | 1.0000000   | 0.53185877  |
5. Conclusion
In this paper, we proposed a novel method to discover the business structure hidden in the IDC room by applying correlation matrix algorithm on the collected monitoring cpu usage dataset. The method is demonstrated on a real world situation, and a case study is launched to evaluate the method, which received a very positive feedback.

Acknowledgments
This work was supported by "Research on Lightweight Active Immune Technology for Electric Power Supervisory Control System", a science and technology project of State Grid Co., Ltd in 2019.

References
[1] Leskovec J and Sosič R. (2016) SNAP. ACM Trans. Intell. Syst. Technol, 8: 1–20.
[2] De Domenico M, Porter M A and Arenas A. (2015) MuxViz: a tool for multilayer analysis and visualization of networks. Journal of Complex Networks, 3: 159–76.
[3] Alkasem A, Liu H, Zuo D and Algarash B. (2018) Cloud Computing: A model Construct of Real-Time Monitoring for Big Dataset Analytics Using Apache Spark. J. Phys.: Conf. Ser. 933, 012018.
[4] Stefanov K S and Gradskov A A. (2016) Analysis of CPU Usage Data Properties and their possible impact on Performance Monitoring. JSFI. 3 (4)