The Design of Voice Over IP Monitoring System

Jaroonsak PHUPHONG*, Chanankorn JANDAENG

Management of Information Technology Program, School of Informatics, Walailak University, Nakhon Si Thammarat, Thailand

(*Corresponding author's e-mail: jaroonrak.phu@gmail.com)

Abstract

Voice over IP (VoIP) is the internet telephone that has been used traditionally as telephone system replacement because it sends voice via Internet/intranet technology, records voice and conversation as it can reduce cost comparing with landline telephone. Because VoIP needs internet/intranet as the main medium to send data, VoIP still effects from the impact of network failure, network low-speed, or security attack. The network monitoring is the tools to monitor and detect the anomaly network behavior. However, most of network monitoring are general and designed for traditional network. They cannot monitor VoIP function directly. This paper proposes the software architecture for VoIP Monitoring. Our system aim is flexibility and scalability. This architecture provides the way to custom in order to implement monitoring system for any organisation. Moreover, it can be applied to other applications which support SNMP protocol and generate audit log to readable text. Network administrator can reformat all inputs to JSON to store and index with ElasticSearch database.

Keywords: VoIP monitoring system; ElasticSearch; Computer network; Simple network management protocol

Introduction

Voice over IP (VoIP) is the internet telephone that has been used traditionally as telephone system replacement. The outstanding of VoIP system is the function of telephone usage, voice and conversation recording. Using VoIP reduces the cost of using landline including telephone call between branches and using telephone call in both domestic and international, etc. Moreover, the security is crucial issues. Network administrator needs to pay attention as well, if the service or infrastructure function has been terminated. Network administrator must analyse the cause and resolves the problem immediately.

The traditional system for VoIP is *Elastix* system [1]. It is the open source software that has been developed from *Asterisk* system by adding more complex hardware and software support to *Elastix* system. However, the *Elastix* system

is not a monitoring function. All activities in the system are recorded with audit log only. On the other hand, the problem of computer network and VoIP system are too complicated. Network administrator cannot deeply analyse the cause of problem in a short time.

The simple method of log analysis is manually analysis with command such as *awk*, *sed*, *tail*, *more*, *grep* etc. Data filtering, report generator and data relation analysis suites for small data. On the other hand, network traffic of VoIP service is too large than manually management. VoIP administrator needs the advanced tools to support this function.

The network monitoring system (NMS) monitors network service and system, resource capacity plan, statistics and accounting, fault management and performance; such as throughput, latency and round trip time. Furthermore, the network monitoring system supports the network under service level agreement (SLA) and network policy [2]. There are many tools to present to monitor audit log and network traffic such as *MRGT* [3], *GrayLog* [4], *Nagios* [5] or *Cacti* [6] etc. Although the various systems can monitor some functions, network administrator still can either solve the problem nor meet the needs directly. Moreover, main function of most network monitoring system specific in network traffic to show and judge network behavior. The specific software for monitor an application is hard to find, especially VoIP monitoring [7].

This paper is the design the software framework for Voice over IP monitoring system that monitor *Elastix* system and all network devices in VoIP System. All devices are enabled SNMP module or installed SNMP agent to reply SNMP request to monitoring system. This paper evaluates our software architecture by implementing in testbed.

Materials and methods

VoIP Architecture

The constraint of our design is the data collection from all components in VoIP ecosystem as shown in Figure 1. The system architecture of VoIP ecosystem consists of VoIP Server, VoIP Gateway, VoIP client and network infrastructure.

The VoIP Server is the core of this services. The traditional open source is *Elastix* system and MySQL as relational database system. The main functions of *Elastix* server are call connection management, log all calling events, simple monitor calling system. The VoIP Client maybe VoIP Phone device or computer installed VoIP application such as *Zoiper*. The client is end node that connected to VoIP server and other devices via internal ID. Whereas VoIP Gateway is the intermediate node that interface between VoIP and Phone to redirect calling connection to public switched telephone network (PSTN).

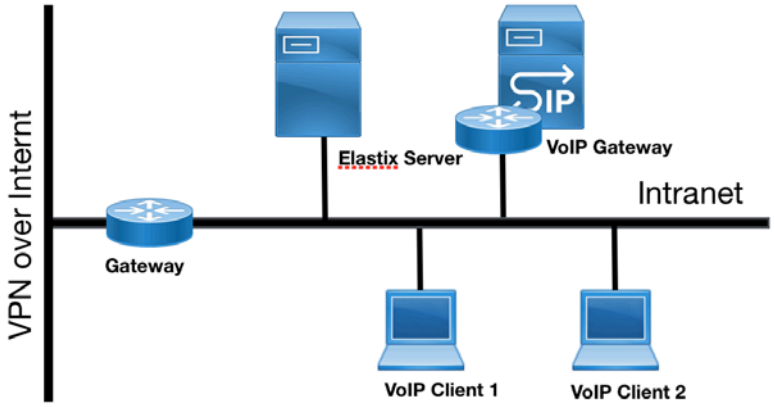


Figure 1. VoIP Architecture

Moreover, the network infrastructure is an important module of VoIP network. All calling data are encapsulated in IP datagram and transits on IP network. Thus all network devices such as router, switches, firewall and wireless access point need to be monitored. If network infrastructure has highly downtime, it will mainly impact to VoIP services. Thus, both VoIP System and network instructor should be monitored in the same time. The security is an issue to be concerned. Firewall device is the main protection from external intruder, while firewall software installing in VoIP Server protects inside intruder or computer infected malware in the organization.

Architectural Overview

Network Monitoring System monitors and analyze data collected from sensor devices. The data format and data source depends on objective of monitoring system. Some data easy to collect directly from devices via monitor agent such as SNMP agent, WMI agent and proprietary agent. On the other hand, some data needs to gather with programming technique. Moreover, the monitoring system analyzes data collecting from event from application and operating system as log file. Most collected data is unstructured data. Thus, it needs to be re-formatted to structured data that is easy to search and analyze in the other step. Structured data are stored and indexed in RDBMS or NoSQL database depends on visualization module in monitoring application.

The system consists of 2 modules: information gathering and VoIPMon. In information gathering module, this module collects data from network infrastructure, operating system and Elastix service. All raw data is collected via Simple Network Monitoring Protocol (SNMP) over TCP/IP [8]. The VoIPMon consists of three functions: data store, dashboard and notification. In data store function, this research proposes key-pair NoSQL database named ElasticSearch. This database is the flexible data structure and all features are indexed that bring to speedy search. The

dashboard and notification are the user interface. The dashboard is web-based application while the notification is e-mail notification. The software architecture is shown in Figure 2 and described below:

1) *Information gathering*: The monitoring system needs 11 factors for VoIP Management: *active calls, processed call, bridged channels, and channel in used, hrProcessLoad, hrStorageSize, hrStorageUsed, diskOnWritten, incomingTraffic* and *outgoingTraffic*. Whereas the security monitoring collects and analyses data from log message in kernel. Because of the 11 factors can be retrieved from management information based (MIB) of SNMP protocol. Thus, the SNMP agent is installed in Elastix Server. In addition, the data collector is implemented and installed in hosted OS to collect factors, transform to key-pair value and record in key-pairs in *syslog-ng,* in local host and remote log server. Collecting data for security monitoring, the firewall application named IP Tables monitors all incoming/outgoing traffic based on firewall rules. The traffic that conflict with the firewall rules will be blocked and logged to *syslog-ng*. Because the log messages of IP tables are represented in key-paired message, they are recorded to *syslog-ng* directly. For network devices, there are two methods for data collecting: 1) if they support SNMP, enable SNMP agent in their devices and reply SNMP response to the collector in VoIPMon and 2) enable logging in remote log server.

2) *VoIPMon*: The main function is storing log messages in database. Raw data is retrieved from devices in VoIP network as log message. Log messages from SNMP agent are transformed to key-paired message and relay to remote log sever that implanting with *syslog-ng*. Whereas, log message from IP Tables already forms in key-paired message. Log message from network devices is stored as unstructured text in *syslog-ng*. This architecture suggests NoSQL database. ElasticSearch is selected because main goal of proposed monitoring system is display information in dashboard. All data write to database only once. All fields are indexed to increase performance in query time. Each record in ElasticSearch is represented in JSON format. Thus key-paired message is transformed to JSON message with *Grok* pattern. *Logstash* is application supporting data preparation phase in the VoIPMon. *Logstash* gathers message from log files and transform to JSON, then store in ElasticSearch database. The importance advantage of *ElasticSeach* is supporting distributed data stores that bring to scalability of VoIPMon to support huge log message. Log messages in JSON format are shown and represented as varieties of chart that depends on user need. *Kibana* is a software in ELK Technology that provide engine to illustrate charts. Indexed data in ElasticSearch is gathered and shown as chart. Thus each chart is combined and shown in dashboard. The last feature of proposed monitoring system is event notification. User configures information such as IP Address, parameter, threshold and reply email to *Logstash*. When value of parameter matches the condition in threshold, notify message will be sent to email.

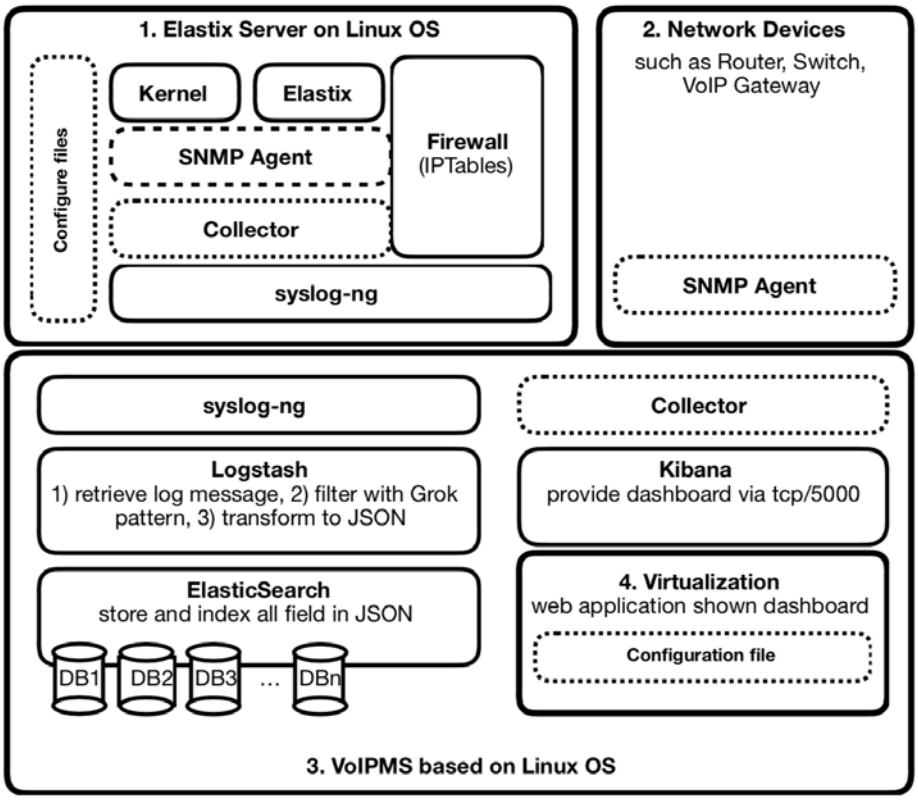


Figure 2. *The software architecture of VoIP Monitoring System*

Data Model

All events are collected via syslog-ng service. After that it is extracted to JSON data with pattern matching technique. The process is shown in Figure 3.

First, collector module periodically requests data from MIB via SNMP protocol, generates result as log message, and forwards to *syslog-ng* as log message as shown in Figure 3. SNMP Log Message shows timestamps, server name, user and log message as representing in key-paired value. After that, *Logstash* read log message and filter with Grok pattern. The result is JSON format as shown in Figure. The JSON consists of data type, label as shown in chart, OID, Accumulative value, snmpagent as Server name, gathering time and current value. Moreover, log message from IPTables is represented in key-paired value too. The example of log message is shown below:

"*Feb 3 01:30:26 192.168.1.16 kernel: PING-DROPIN=eth0 OUT= MAC=08:00:27:bf:56:de:f0:79:59:8f:14:c5:08:00 SRC=192.168.1.100 DST=192.168.1.16 LEN=65528 TOS=0x00 PREC=0x00 TTL=128 ID=522 PROTO=ICMP TYPE=8 CODE=0 ID=2 SEQ=47476*".

This message is processed like SNMP Log message with difference Grok pattern and send the source and destination address, protocol, and timestamp to *ElasticSearch.*

Scalability

The importance issue of network monitoring is scalability. This software architecture aware in this issue. Because log messages are generated very second. All log messages are stored in database. Capability of storage are increased rapidly. It impacts performance of operating system in term of increasing of disk access time. Moreover, it impact dashboard's performance because ElasticSearch application supports distributed storage, then adding ElasticSearch server is the good way to solve this problem. In addition, scalable feature of ElasticSearch increase reliable of data because all data is redundancy. When any database server fails, others can continue to store data and show dashboard.

SNMP Log Message

```
Mar 11 10:50:01 ubuntu root: data=snmp snmp_type=stat
label=CallsProcessed oid=1.3.6.1.4.1.22736.1.2.6.0
accvalue=Counter32: 100 host=192.168.1.16
timestamp=1520740201 value=0
```

Grok Pattern

```
%{DATA:data} snmp_type=%{DATA:snmp_type} label=%{DATA:label}
oid=%{DATA:oid} accvalue=Counter32: %{BASE10NUM:accvalue:int}
host=%{IP:snmpagent} timestamp=%{DATA:timetick:int} value=%
{BASE10NUM:value:int}
```

JSON Data

```
{
   "data": "snmp",
   "snmp_type": "stat",
   "label": "CallsProcessed",
   "oid": "1.3.6.1.4.1.22736.1.2.6.0",
   "accvalue": "100",
   "snmpagent": "192.168.1.16",
   "timestamp": " Mar 11 10:50:01 ",
   "value": “0"
}
```

Figure 3. *Transformation from log message to JSON: SNMP Log Message*

Flexibility

There are no network monitoring system suites for all situation and organiZation. The commercial network monitoring systems are fixed the number and type of function. When there is a new requirement, user maybe pay for

more function or the vendor rejects the request. The proposed architecture is designed base on flexibility of system administrator. All device and server are monitored by system administrator, whereas the new dashboard can be generated and add to web site via configuration file. Thus, this system is very flexible for system administrator.

Results

Implementation

We implement the proposed architecture and test in the testbed that consist of VoIP Server, VoIP Client and VoIPMon as shown in Figure 4. The VoIP Server and VoIPMon are personal computers installed Linux OS. The hardware specification is Intel core i5 2.4GHz, 8 GB main memory, 160 GB Hard disk and 1 Gb Ethernet Card. Whereas, VoIP client installs VoIP software named Zoiper. Our testbed is composed of eight clients. Each client is MS Window 10 home edition installing in virtual machine. The virtual machine specification is 1 vCPU, 1024 GB main memory, 64 GB hard disk and 1 GB vNIC.

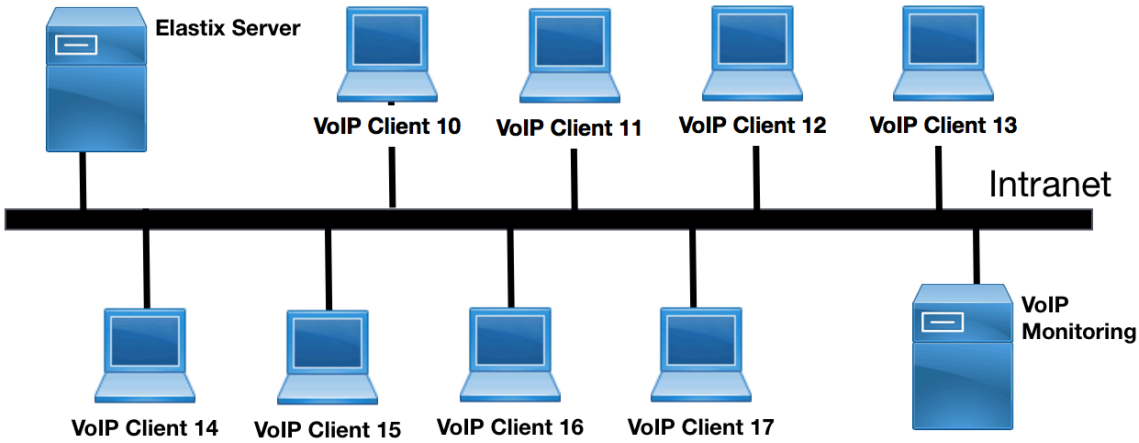


Figure 4. *Testbed of VoIP System*

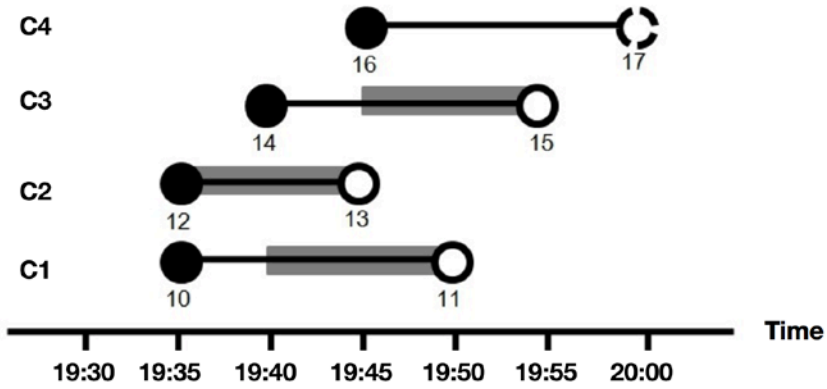
Performance Evaluation of VoIP with VoIPMon

This research evaluates VoIP performance in 4 metrics: Active Calls (AC), Bridged Channel (BC), Channels In Used (CU), Calls Processes (CP) with 4 case studies.

- *Active calls* is the number of call pair in duration time. This factor indicates the concurrent calling in VoIP system. If this metric high means that there are many concurrent calling.
- *Bridged channel* is the number of successful calling. This indicator will be counted when the connection is completed. Active call and Bridged channel are difference. The active call counts all connections. However, Bridged channel count only completed connection.

- *Channels in Used* is number of calling in duration time including miscall connection.
- *Calls Processes* is the number of Elastix process. This metric is accumulative value until VoIP server is reset.

VoIP administrator need to monitor server performance in term of CPU usage, Memory usage and Network Traffic in order to plan and manage VoIP system. If each resource is consumed in higher level than threshold, VoIP administrator needs to re-plan the resources usage before VoIP system fails. The Figure 5 shows the calling simulation that consists of four situations: three success calling and one miscall. The experiential result is shown in Figure 6.



Case	Source	Destination	Start Time	Answering Time	Hang up Time	Duration calls (min)
C1	#10	#11	19:35	19:40	19:50	10
C2	#12	#14	19:35	19:35	19:45	10
C3	#14	#16	19:40	19:45	19:55	10
C4	#16	#18	19:45	Miscall	N/A	N/A

Figure 5. *The 4 test case situations*

Results

Figure 6 shows chart of 4 metrics: Active Calls, Bridged Channel, Channels In Used and Calls Processes. Each chart is the line chart: x-axis represents interval time every 30 minutes while y-axis represents each metrics.

Node#10 and #12 calls to node#11 and #13 at 19:35. After that Node#14 and #16 calls to participant at 19:40 and 19:45. The result shows that AC start from two nodes at 19:36 because network monitoring polls every a minute and there are two node called to the participant. After that, Node#14 and #16 called their call pairs. AC is three nodes because connection C1 stops. AC continuously decrease at 19:50 and 19:55. Finally, AC is set to 0 at 20:00 because all connection hangs up. Channels in Used is number of successfully communication node that it does not include miscall. From the experimental we found that the CU is three call pairs, thus, the CU counts all online node. Call

Processed is accumulative value; thus CP is increased when client node calls the participant. This indicates the counting of activated node in VoIP system.

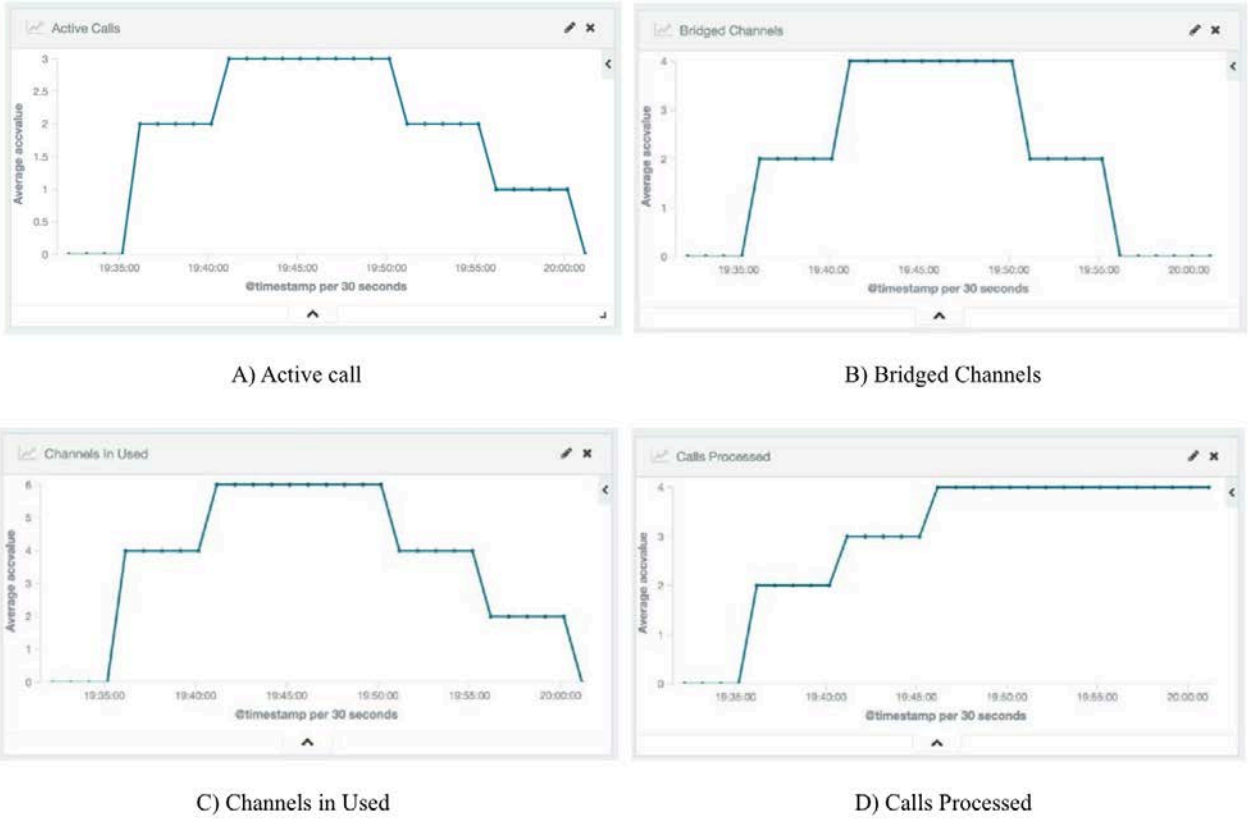


Figure 6. *Example of experimental result*

Performance Analysis

Storage is an importance metric for implementation of network monitoring system. Normally, all events are recorded in log system as plain text or text file or semi-structure. The log message is readability text, network administrator can manually analyze their plain text. This experiment transforms semi-structure message to structure message as JSON in order increase flexibility of data analysis and retrieve. The four call pairs randomly call for 12 hours. The experiment shows storage usage in Figure 7.

Figure 7 shows that x-axis is duration time while y-axis is used storage size. We compare between log messages representing in plain text in log file and JSON in ElasticSearch. We found that there are duplication messages of plain text in JSON object that result in ElasticSearch consumes storage more than plain text in system log. When we extract the overhead from JSON we found that storage size of ElasticSearch is increased. By average, ElasticSearch consumes storage about 1.91 MB per hour.

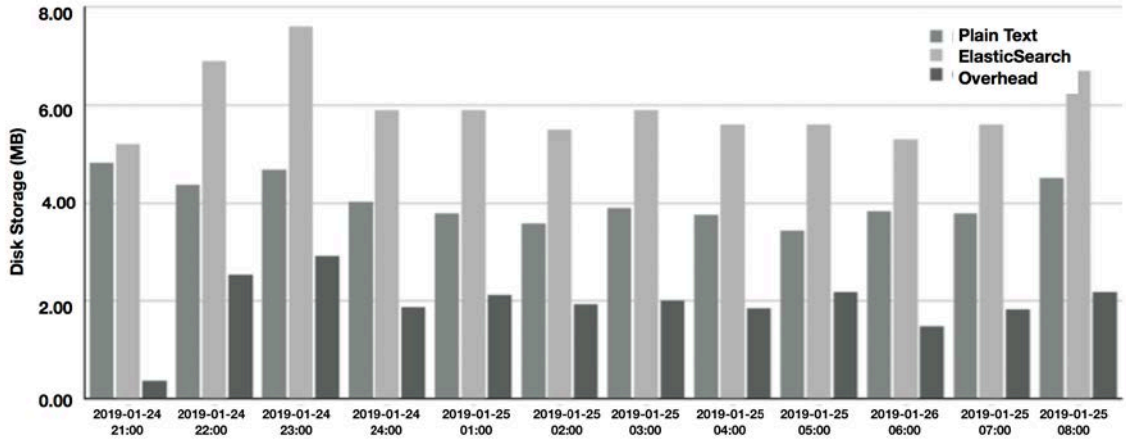


Figure 7. *Comparison of Disk Usage between Audit Log and ElasticSearch*

Conclusions

This research proposed the software architecture of VoIP Monitoring System called VoIPMon. This monitoring collects data from log files that generated by SNMP, Firewall and other data sources. This architecture provides data transformation module to transform non-structure/semi-structure to structure data and store in NoSQL databased named *ElaticSearch*. All fields in databased are indexed and retrieved by *Kibana* to represent information into chart and linked groups of chart as dashboard. The scalability, flexibility and storage consumption are concern. Finally, we evaluate storage consumption. The experimental show that *ElasticsSearch* consumes storage for 1.91 MB per hour.

References

[1] Elastix: Your Linux PBX Unified Communications Solution, Retrieved from: https://www.elastix.org (2019).

[2] M. Subramamian, A. Timothy, N. Gonsalves and U. Rani, “Network Management: Principles and Practice”, Dorling Kindersley, India (2010).

[3] MRTG, Retrieved from: http://oss.oetiker.ch/mrtg/doc/mrtg.en.html (2019).

[4] Graylog: Open Source Log Management, Retrieved from: https://www.graylog.org (2019).

[5] Nagios: Open Source Project, Retrieved from: https://www.nagios.org/ (2019).

[6] Cacti, Retrieved from http://www.cacti.net (2019).

[7] X. Liu, F. Liu, W. Zhou and Y. Xie, “The design and implementation of VoIP flow monitoring system”, Proceeding of 2011 International Conference on Advanced Intelligence and Awareness Internet (AIAI 2011), Shenzhen, 385-388 (2011).

[8] SNMP Reference Guide: Media Gateways, SBCs & MSBRs, Retrieved from http://tinyurl.com/yyczh97y (2019).