USENIX Association

Proceedings of
LISA 2002:
16th Systems Administration
Conference

Philadelphia, Pennsylvania, USA
November 3–8, 2002

MieLog: A Highly Interactive Visual Log Browser Using Information Visualization and Statistical Analysis

Tetsuji Takada & Hideki Koike – University of Electro-Communications

ABSTRACT

System administration has become an increasingly important function, with the fundamental task being the inspection of computer log-files. It is not, however, easy to perform such tasks for two reasons. One is the high recognition load of log contents due to the massive amount of textual data. It is a tedious, time-consuming and often error-prone task to read through them. The other problem is the difficulty in extracting unusual messages from the log. If an administrator does not have the knowledge or experience, he or she cannot readily recognize unusual log messages. To help address these issues, we have developed a highly interactive visual log browser called "MieLog." MieLog uses two techniques for manual log inspection tasks: information visualization and statistical analysis. Information visualization is helpful in reducing the recognition load because it provides an alternative method of interpreting textual information without reading. Statistical analysis enables the extraction of unusual log messages without domain specific knowledge. We will give three examples that illustrate the ability of the MieLog system to isolate unusual messages more easily than before.

Introduction

Administration of computers has become more important than ever because of the increasing role of computers and networks in providing various services to our daily life. It is therefore necessary to conduct them continuously and properly as part of our modern infrastructure.

Computer log inspections are the most fundamental tasks in administration, since most of the events occurring in computers and networks are recorded into log-files. Administrators, therefore, must inspect them periodically. When they find an anomaly in the log-file, they must make an appropriate response as soon as possible. Today's security threats to a computer network increase the importance of log inspections to help detect possible breaches.

Although administrators recognize the importance of log inspection, the task is often not performed regularly at many computer sites. One reason is that it is a tedious and time-consuming task due to the large amount of textual data. Another reason is that it requires skilled knowledge to recognize an unusual message in the log-files.

We have developed a highly interactive log browser, called "MieLog," which uses information visualization and statistical analysis to help alleviate some of the problems involved in log monitoring. The purpose of the system is to assist administrators to inspect computer logs manually. MieLog consists of three main approaches. One is information visualization to improve the recognition load of textual data. Another is a high level of interactivity which makes it easier to filter out or extract information from log data. The last is statistical analysis which provides inspectors with various tools to help detect unusual messages in the log.

This paper is organized as follows: First, we mention the issues and importance of computer log inspections. The next section presents the system overview and the detail of each module of MieLog. Subsequently, we explain the visualization method of computer logs and interactive functions. then we show some examples of computer log inspections using MieLog. Finally, we discuss related work and proposed future enhancements.

Problems of Computer Log Inspections

There is no doubt that log inspections are indispensable for computer administration. Administrators, however, regard them as tedious, time-consuming and often unrewarding tasks. Therefore, although some administrators are aware of the importance of the tasks, they hesitate to perform them. Indeed, a recent security survey in Japan shows that such tasks have not been performed sufficiently even by Internet service providers.

We can define log inspections in more specific terms:

1. Administrators retrieve a log and analyze their contents by reading through the messages.
2. Administrators extract unusual messages from the log.

We will consider each of these problems in further detail. The factors that make it difficult to interpret the log contents are that:

- Log messages are recorded as text.
- Logs usually contain a huge amount of data.

- Logs have various kinds of formats and content.

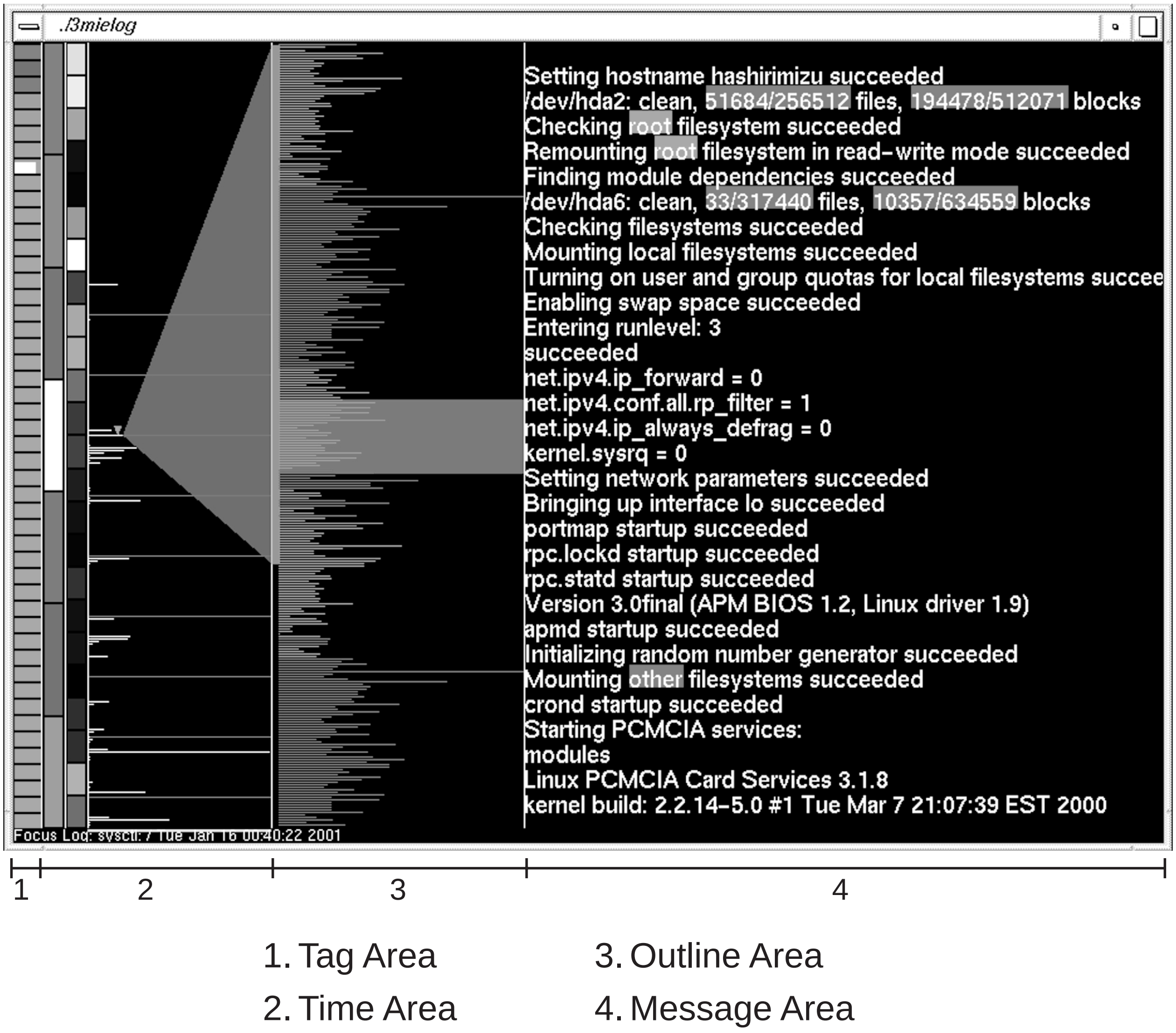
These factors clarify the problems that administrators must face. Administrators must read through the log messages to understand them and they must spend many hours doing so. Thus, it is almost impossible to manually inspect computer logs at a large computer site. Administrators also require specialized knowledge about the logs, since recording formats, contents and existing directories of each log may be completely different. There are many problems with just the log recognition stage in log inspection tasks.

Next, we list the factors that make it difficult to isolate unusual log messages.

- Log messages resulting from a problem or intrusion may be small and buried among other unimportant messages.
- It is difficult to build rules for automatic extraction of all unusual messages.
- There are cases when the administrator cannot determine whether the log message results from an abnormal event or not.

Computer log-files contain various kinds of messages. They contain not only errors and warnings but also operating system status and notice from applications. In general, they contain only a few important messages, while the others are less important messages. Administrators, therefore, must be able to extract the important messages from the log.

The next problem is that it is difficult to formulate the rules for unusual log message extraction. There are two reasons. One is that no administrator knows what constitutes unusual messages for all cases. The other is that the rules are highly dependent on both administrator's knowledge and the environment of the site. We confer that it is important to build rules for extracting known problem log messages. We believe that it is also important to inspect the logs



```
Setting hostname hashirimizu succeeded
/dev/hda2: clean, 51684/256512 files, 194478/512071 blocks
Checking root filesystem succeeded
Remounting root filesystem in read-write mode succeeded
Finding module dependencies succeeded
/dev/hda6: clean, 33/317440 files, 10357/634559 blocks
Checking filesystems succeeded
Mounting local filesystems succeeded
Turning on user and group quotas for local filesystems succee
Enabling swap space succeeded
Entering runlevel: 3
succeeded
net.ipv4.ip_forward = 0
net.ipv4.conf.all.rp_filter = 1
net.ipv4.ip_always_defrag = 0
kernel.sysrq = 0
Setting network parameters succeeded
Bringing up interface lo succeeded
portmap startup succeeded
rpc.lockd startup succeeded
rpc.statd startup succeeded
Version 3.0final (APM BIOS 1.2, Linux driver 1.9)
apmd startup succeeded
Initializing random number generator succeeded
Mounting other filesystems succeeded
crond startup succeeded
Starting PCMCIA services:
modules
Linux PCMCIA Card Services 3.1.8
kernel build: 2.2.14-5.0 #1 Tue Mar 7 21:07:39 EST 2000
Focus Log: sysctl:7 Tue Jan 16 00:40:22 2001
```

Figure 1: A display image of MieLog.

periodically to find unknown unusual log messages and rebuild or refine extraction rules based on the new information.

The last problem is that administrators cannot judge the importance of each log message based solely on one kind of log-file. Because each log message contains only partial information about an event that has occurred in a computer, a more reliable judgment requires log messages from multiple log-files. It is therefore necessary to collect other related information from various log-files and analyze them comprehensively. This would require many operations, as well as extensive knowledge and time.

In this paper, we propose the use of information visualization and statistical analysis to address the above problems.

In general, the number of unusual log messages is small in typical log-files. If we obtain frequency information from a log using statistical analysis, it is possible to isolate such log messages. It helps administrators to find truly unusual log messages. Furthermore, MieLog visualizes frequency information and the log file itself as a figure. It reduces the recognition load of log messages when inspecting them. The reason is that the method of log message recognition changes from ‘‘reading’’ to ‘‘looking.’’

MieLog does not just visualize a log as a figure. It also adds a high level of interactivity. Many of the interactive functions help perform filtering of log messages in various ways. An inspector can execute commands by direct interaction with a visualized figure. The combination of the two features makes it possible to reduce the problems of inspecting logs by humans.

MieLog: System Overview and its Visualization

We developed an interactive log information browser called ‘‘MieLog’’ based on the considerations presented in the previous section. In this section, we describe the system modules of MieLog and the features which address the above mentioned problems.

We used C++ programming language and OpenGL library in development of MieLog. The visual

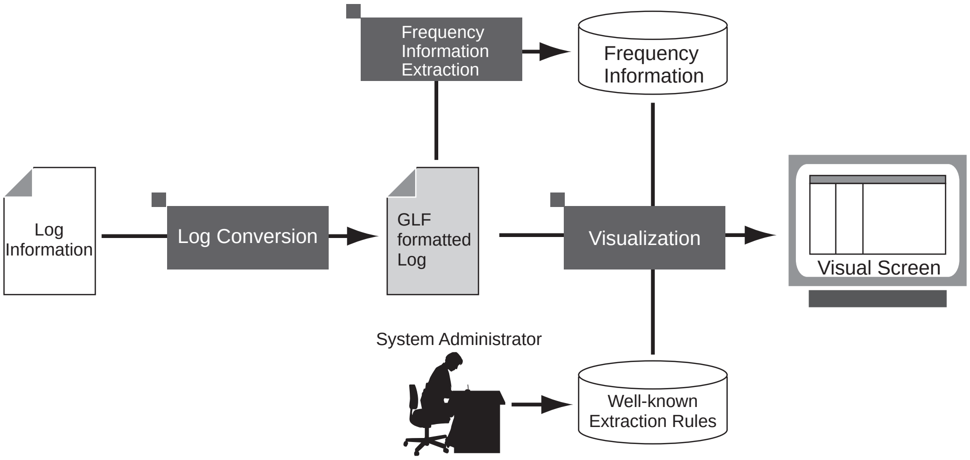


Figure 2: A process module overview of MieLog.

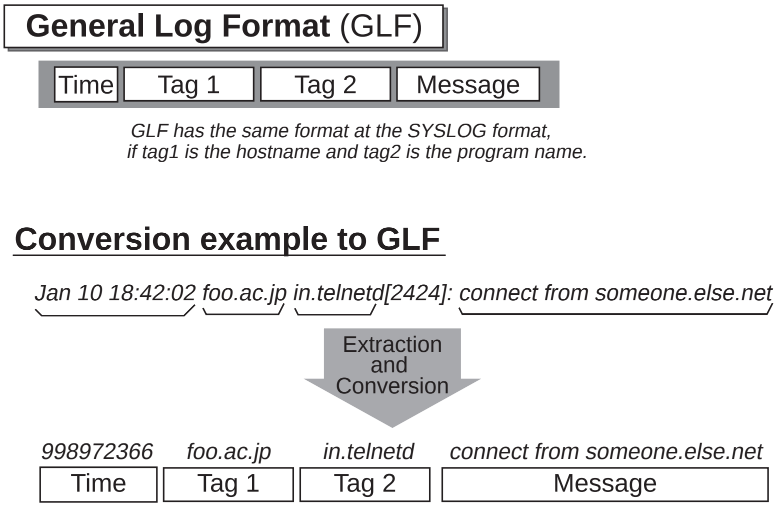

General Log Format (GLF)
Time
Tag 1
Tag 2
Message
GLF has the same format at the SYSLOG format,
if tag1 is the hostname and tag2 is the program name.
Conversion example to GLF
Jan 10 18:42:02 foo.ac.jp in.telnetd[2424]: connect from someone.else.net
Extraction and Conversion
998972366
foo.ac.jp
in.telnetd
connect from someone.else.net
Time
Tag 1
Tag 2
Message

Figure 3: Specification of general log format and conversion example.

screen of MieLog is shown in Figure 1. The screen is composed of four visualization areas. They are the ''Tag area,'' the ''Time area,'' the ''Outline area'' and the ''Message area'' respectively in order from left to right of the screen. The left three areas visualize different ''characteristics'' of the log, while the fourth area is for viewing the actual message text. MieLog also visualizes the relationship between each area clearly.

The system modules of MieLog are illustrated in Figure 2. MieLog is composed of three modules. We explain the details of each module in the following sections.

Log Conversion Module

This module converts logs with various recording formats into an intermediate format called the ''Generalized Log Format'' (GLF).

There are various problems involved in inspecting a log. One is the different recording formats used and the types of contents of each log message. The other is that administrators must have extensive knowledge about the log: where the log-file exists, how to get the log messages and which log-files should be inspected. To simplify these problems, we provide two types of tools. One is the log collection and conversion tool. The other tool is for merging converted logs. Figure 3 shows the syntax of the ''Generalized Log Format'' and a conversion example.

As you can see, the General Log Format consists of four elements. The type of each element is a character string except for the time element, which is an integer value in seconds. This format is very similar to the message format of the syslog daemon on UNIX systems.

This module contributes to a couple of advantages. MieLog has the ability to browse through log messages recorded in several log-files at one time because the conversion of log message enables the integration of various computer logs into one. The integration of logs are based on the recorded time stamp of each message. These functions reduce the number of operations and time involved when administrators have to inspect several logs. Moreover, this reduces the difficulty of the comprehensive judgment because an inspector receives the time correlation of log messages.

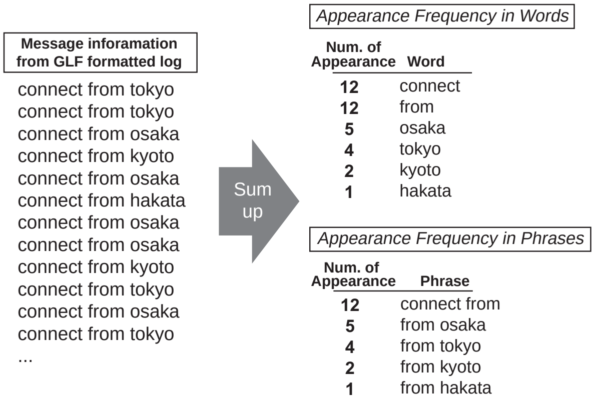
The other role of this module is pre-processing for statistical analysis in order to extract frequency information from the log.

Frequency Information Extraction Module

This module extracts frequency information from GLF formatted log messages using statistical analysis. MieLog uses this information to help extract unusual log messages without pre-defined keywords. This approach is based on the following concept: even if a log has a massive amount of message data, there are generally only a few key messages. In other words, using such information, we can extract at least the ''candidates'' of unusual log messages. This assists an inspector in recognizing anomalous messages even if administrators have no prior knowledge or experience about the log.

We explain how to extract frequency information with respect to each element in the GLF as follows:

- **Frequency information regarding the time**. There are two types of frequency information extracted from the time element of the GLF. One is the number of log messages that occur in each unit of time in a periodical time span. The other is the number of log messages in each unit of time for the entire period of the log.
- **Frequency information regarding the tag**. This module counts the number of appearances of each tag and keeps them sorted in descending order.
- **Frequency information regarding the message.**. We focus on a word and a phrase in log messages as a unit of the analysis. A phrase in MieLog is defined as a series of two words in a message. The module counts the number of appearances of them (Figure 4).



Message inforamation from GLF formatted log

```
connect from tokyo
connect from tokyo
connect from osaka
connect from kyoto
connect from osaka
connect from hakata
connect from osaka
connect from osaka
connect from kyoto
connect from tokyo
connect from osaka
connect from tokyo
...
```

Appearance Frequency in Words

Num. of Appearance	Word
12	connect
12	from
5	osaka
4	tokyo
2	kyoto
1	hakata

Appearance Frequency in Phrases

Num. of Appearance	Phrase
12	connect from
5	from osaka
4	from tokyo
2	from kyoto
1	from hakata

Figure 4: Feature extraction of messages.

Using MieLog, it is also possible to extract unusual log messages using keywords that an inspector already knows. When an inspector defines keywords, MieLog highlights these words or phrases visually.

Information Visualization Module

MieLog visualizes log messages by combining three kinds of sources: GLF-formatted log messages, frequency information and pre-defined keywords. This module also makes MieLog a highly interactive system. MieLog has a variety of interactive functions to help extract unusual messages. Visualization provides the inspectors with the following two advantages: One is to reduce the load on recognizing a textual message. The other is that it enables administrators to introduce human decision making into judging whether each message seems to be unusual or not.

Visual Representations and Interactive Functions

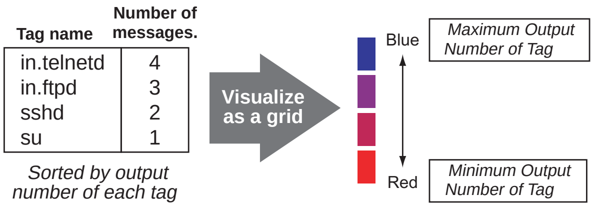
In this section, we describe the visualization method and interactive functions of MieLog.

Visual Representation of MieLog

The information visualization module creates a visual screen such the one shown in Figure 1. The screen of MieLog is composed of four visual areas as mentioned in the previous section. We will explain the visual representation of each area in this section.

Tag area

The tag area visualizes frequency information of tags as a vertical grid (Figure 5). Each colored tile in the grid represents a corresponding tag information. The number of tiles represents the total number of tags.



Tag name	Number of messages.
in.telnetd	4
in.ftpd	3
sshd	2
su	1

Sorted by output number of each tag
Visualize as a grid
Blue
Red
Maximum Output Number of Tag
Minimum Output Number of Tag

Different tiles represent different tags.
The color of the tiles represents
the number of times the tags appeared.

Figure 5: Visualization method of tags in log.

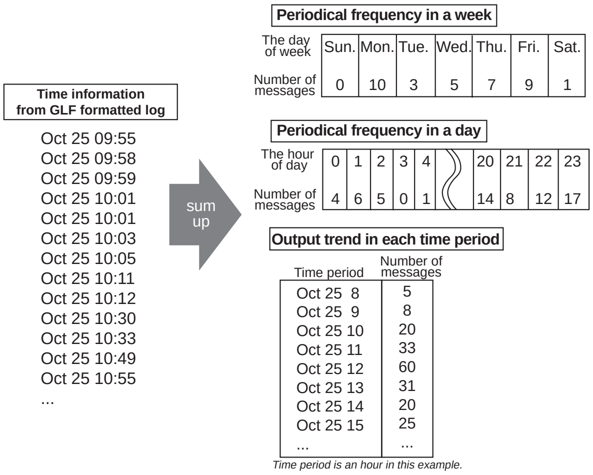
The color of the tiles represents the value of frequency information of each tag. A blue tile indicates that the corresponding tag has the highest frequency value in the log, while a red tile indicates that the corresponding tag has the lowest frequency value. Other tiles with intermediate frequency values have intermediate colors between red and blue. This visualization makes it possible to understand the number and frequency of each tag. The name of each tag is displayed at the bottom of the screen.

MieLog uses another coloring scheme in this area based on the number of tiles, namely the number of tags. The color of each tile is evenly gradated from blue to red. It is easier to distinguish each tile than the colors based on the frequency values.

Time area

The time area is subdivided into three areas. The right-most column of the time area shows a histogram. The time is assigned from top to bottom, and the value axis is assigned from left to right. It shows how many messages area produced in each unit span. The two left columns in the time area represent the appearance frequency information in different periodic time divisions. As you can see, the left grid has seven tiles and the right has twenty four tiles. This indicates that the left grid represents the appearance frequency information in a week and the right represents them in a day. The representation method of these grids is the same as that of the tag area except for the coloring. The coloring of this area is a gradation between white and black instead of blue and red (Figure 6).

These visual representations make it easier to recognize time-characteristics of the log.



```
Oct 25 09:55
Oct 25 09:58
Oct 25 09:59
Oct 25 10:01
Oct 25 10:01
Oct 25 10:03
Oct 25 10:05
Oct 25 10:11
Oct 25 10:12
Oct 25 10:30
Oct 25 10:33
Oct 25 10:49
Oct 25 10:55
...
```

Periodical frequency in a week

The day of week	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.
Number of messages	0	10	3	5	7	9	1

Periodical frequency in a day

The hour of day	0	1	2	3	4	20	21	22	23
Number of messages	4	6	5	0	1	14	8	12	17

Output trend in each time period

Time period	Number of messages
Oct 25 8	5
Oct 25 9	8
Oct 25 10	20
Oct 25 11	33
Oct 25 12	60
Oct 25 13	31
Oct 25 14	20
Oct 25 15	25
...	...

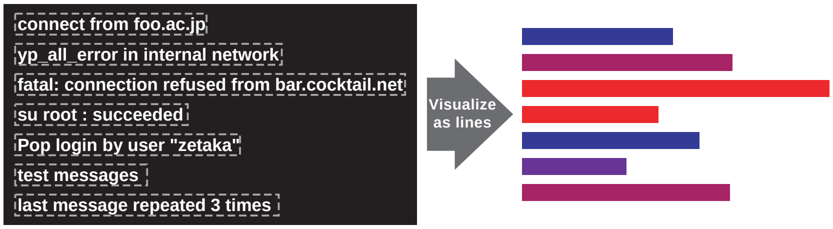
Time period is an hour in this example.

Figure 6: Visualization method of time in logs.

Outline area

This area displays the outline of log messages. Each log message is represented as a colored line. The length of the line is the string length of the log message. The colors of the lines are the same as the grid's colors in the tag area. In other words, the color of each line is assigned to the color of the corresponding tag defined in the tag area (Figure 7).

This visual representation enables administrators to recognize log messages as a visual pattern based on the length and the frequency with which they appear. As a result, it is possible to browse many log messages



```
connect from foo.ac.jp
yp_all_error in internal network
fatal: connection refused from bar.cocktail.net
su root : succeeded
Pop login by user "zetaka"
test messages
last message repeated 3 times
```

The outline area visualizes the log messages
as colored lines corresponding to the message length.

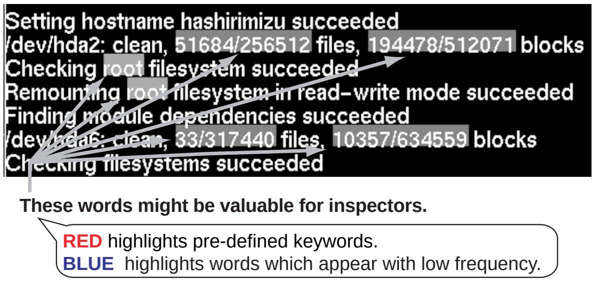
Figure 7: Visualization method of outlines of log messages.

at once, unlike textual representations. Since the line color depends on an appearance frequency of the tag, administrators can also judge whether each message appears to be unusual or not. These provide inspectors with the opportunity to pinpoint unusual log messages before they read the textual log messages.

In the center of this area, a transparent square exists, representing the correlating section between the outline area and the message area. The region of the highlighted square in the outline area is the section displayed in the message area.

Message area

The message area represents actual textual log messages. It is a view similar to that of a text editor, with the exception that it has highlighted words or phrases. Words and phrases highlighted in red and blue (Figure 8). The words and phrases highlighted with red represent the keywords specified in the predefined keywords. Those highlighted in blue represent words with a low appearance frequency value. An inspector must define the threshold value manually if he or she wants to extract the words and phrases with a low appearance frequency value. These features make it possible to extract not only known key messages but also potentially suspect messages.



```
Setting hostname hashirimizu succeeded
/dev/hda2: clean, 51684/256512 files, 194478/512071 blocks
Checking root filesystem succeeded
Remounting root filesystem in read-write mode succeeded
Finding module dependencies succeeded
/dev/hda6: clean, 33/317440 files, 10357/634559 blocks
Checking filesystems succeeded
```

These words might be valuable for inspectors.

RED highlights pre-defined keywords.
BLUE highlights words which appear with low frequency.

Figure 8: Visualization method of log messages and its features.

Interactive Functions

MieLog has a variety of interactive functions that perform various filtering of log messages. Using these functions, administrators can extract log messages using various visual transformations. This capability effectively assists inspectors by extracting the messages that meet a specific pattern. In this section, we describe the interactive functions of MieLog. We explain them with respect to each visual area.

Tag area

An interactive function in the tag area allows the extraction of log messages with a specific tag. If administrators want to filter log messages using tag information, they simply specify their focused tag by clicking the tile in the grid with the mouse. They then obtain a new visual screen that displays only the log messages with the specified tag.

It is possible for inspectors to specify not only one tag but also multiple tags in the filtering. It is also possible to specify the tags based on appearance frequency information.

Time area

An interactive function in the time area extracts the log messages based on their recorded time. There are two types of visualization methods in this area: grid visualization and histogram visualization. We explain the interactive function of each of them respectively.

In grid visualization, an inspector can extract log messages based on two types of periodical time spans. They are the hour of the day and the days of the week. The method of filtering, as in the tag area, is to click the tile with the mouse. The administrator can specify multiple tags in a grid. They can also specify multiple tags in two separate grids. In such a case, MieLog extracts the log messages based on the "AND" condition in each time span. In other words, it is possible to extract the log messages that were recorded in 18, 19 and 20 o'clock on Saturday and Sunday just by clicking the five tiles.

In histogram visualization, an inspector can extract the log messages based on the number of log messages in each time span. The filtering method is described as the following. First, the inspectors should select the type of the filtering. There are three filtering conditions: "less than," "nearly equal," and "more than" a threshold value based on the number of log messages in each time span. Next, the inspector should define a threshold value. A vertical line is drawn when the inspector drags the mouse pointer by pushing the right button in the histogram area. That line represents the threshold value for filtering. The inspector can define the threshold value interactively using the visual representation. Finally, the inspector releases the right mouse button to fix the threshold value according to the location of the mouse pointer. The filtering process, then, starts running using the threshold value and the previously defined filtering mode. A visual representation reflects the filtered result.

Outline area

An interactive function in the outline area enables inspectors to extract log messages based on the length of the log message.

Whenever the inspector defines a base length for filtering by manipulating a mouse, they obtain a new visual screen that displays only the messages with a certain length. There are three filtering conditions. The first filtering condition extracts messages shorter than the base length. The second condition extracts messages which are nearly equal to the base length. The third condition extracts messages longer than the base length. The method of filtering is the same as the filtering method based on the output number in the histogram of the time area.

The outline area has another interactive function which enables direct access to the specific log message for a detailed look, displaying it in the message area. The number of visualized log messages is much greater in the outline area than in the message area. Therefore, many log messages visualized in the outline area are not visible in the message area. If unusual log messages seem to exist in the log messages, it is natural that administrators would want to know the details of them. This interactive function helps them to inspect such log messages more easily.

Message area

An interactive function in the message area extracts log messages that include specific words or phrases. In other words, it is possible to filter log messages by a word or a phrase. The method is described as follows.

First, the inspector should choose words or phrases in the menu. This operation is to decide the unit element for filtering. Second, a filtering condition should be selected. There are three filtering conditions: ''and,'' ''or,'' and ''not.'' These filtering conditions represent the logical relation between selected words or phrases. Filtering with ''and'' condition extracts the log messages that include all selected words or phrases. Filtering with ''or'' condition extracts the log messages that include at least one of the selected words or phrases. The above two filtering conditions become effective when an inspector selects more than one word or phrase. Filtering with ''not'' condition extracts the log messages that do not include the selected words or phrases. It is also possible for an inspector to use this filtering condition when he or she selects only one word or phrase.

The ''not'' filtering is extremely useful to reduce the amount of visualized messages because it enables the inspector to erase some messages from the inspection target. Using this filtering, the administrators filter out the well-known (i.e., useless) log messages step by step. This function assists the administrator to narrow the inspection target easily and interactively.

There are other kinds of interactive functions that are not closely related to other visual areas.

Defining Keywords and Key Phrases

When the inspector defines keywords or key phrases that are already known as unusual log messages, MieLog highlights them in a manner that the inspector can easily recognize the existence of them. The definition of keywords and key phrases is usually done using another tool such as a text editor. It is, however, possible to define them using MieLog itself through the GUI.

There is two methods of defining keywords or key phrases in MieLog. One is to input them through the GUI. The other method is described as follows. First, the inspector selects a word or a phrase as a keyword or a key phrase on the screen of MieLog using the mouse. Next, he or she starts running the GUI for keyword definition. Then, selected words or phrases are automatically input in the GUI. If the inspector pushes the ''OK'' button on the GUI, they are defined as a keywords or key phrases. Namely, the latter method provides a convenient method of input of keywords or key phrases for the inspectors.

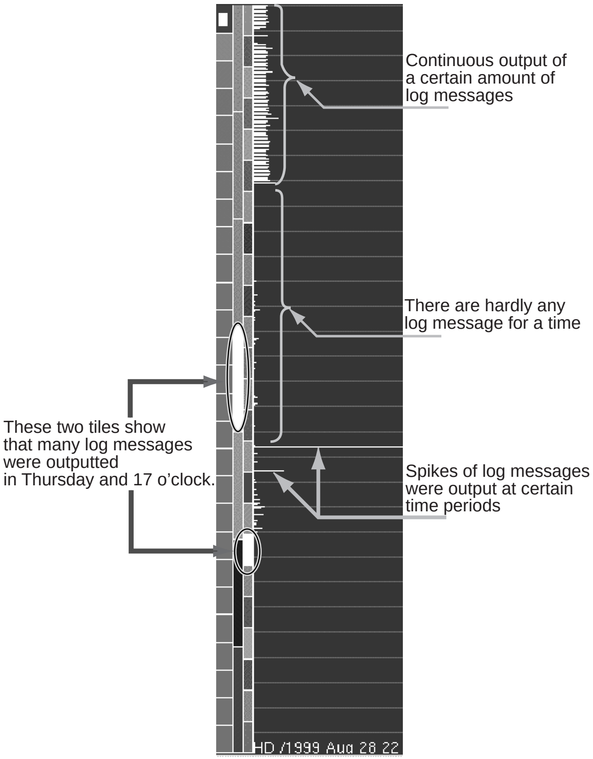


Figure 9: An example of an investigation focusing on time.

Summarizing Log Messages

MieLog has a log summarization function that eliminates the duplication of log messages. Computer log-files generally contain massive amounts of messages. One of the reasons is that relatively unimportant messages are recorded repeatedly in the log. These messages are usually a result of a proper event of the operating system or applications. MieLog, therefore, enables the inspector to summarize log messages interactively in order to reduce the inspection target. If the inspector executes this function, all visualized log messages in MieLog become unique. This makes it possible to reduce the number of log messages and avoid redundant log inspection.

Log Inspection Examples Using MieLog

In this section, we show three examples which demonstrate how to find log messages that seem to show an abnormal behavior using MieLog. We also explain how to use the interactive functions for effective browsing in each case.

An Inspection Example using Log Recording Time Visualization

Figure 9 shows an example of visual representation of the time area.

There are two grids in the time area. Each grid has one bright white tile. This visualization shows that many log messages were recorded at a certain hour of the day and a certain day of the week. In this example, these are 17 o'clock and Thursday. This is a notable indication to help find an unusual log message. Such an indication would be regarded as an abnormal event in many cases. occurrences in many cases. The administrators, therefore, should inspect the log messages recorded in that period of time. It is easy for them to view only the log messages recorded in those periods of time, if they make use of the interactive log message filtering by time. They would simply click the two white tiles with the mouse.

Next, we look at the visualization of an output trend as a histogram in the time area. From the example diagram, it is possible to recognize certain notable activity as listed below:

1. Up until a certain time, a regular number of log messages were continuously recorded in each time span.
2. After this period, no log messages were recorded. This situation continued for a while.
3. Some time later, there are two time spans that recorded large spikes of log messages.

It is possible for the inspectors to recognize all of these indications without actually reading through the log messages.

We propose that there are three specific areas that the administrator should inspect the log in further detail based on the above indications. One is the time when message output was lost. The others are the two time spans when a large number of log messages was recorded.

It is easy to perform these inspections using the interactive function of MieLog. Administrators can easily access the log messages that were recorded in a specific time span just by clicking the lines in the histogram of the time area. They will get a new visual screen that shows the messages recorded in the specific time span only.

In this example, as a result of the above indications, the inspectors was able to determine the following things: The reason why log messages ceased after a certain time is that a system program did not start after the system configuration was modified. And the massive number of log messages generated in specific time spans were a result of running examination of new software by an another administrator. In this case, both indications did not result from an abnormal event.

We have shown an example in which MieLog was able to represent various indications to the inspector just by focusing on the time area. These indications assisted the inspector to find unusual messages through time trends and frequency, which would have been almost impossible to discover reading through conventional text.

An Inspection Example using Log Outline Visualization

Next, we focus on the visual representation of the outline area. Figure 10 shows three visual representations of the outline area. The left visualization seems to have been made during a normal status. The other visualizations seem to indicate an abnormality.

Focusing on the left visualization, There are three characteristics which can be recognized as follows:

1. Most of the log messages have nearly same length.
2. There are the same series of log messages output towards the middle and the bottom of the visualization.

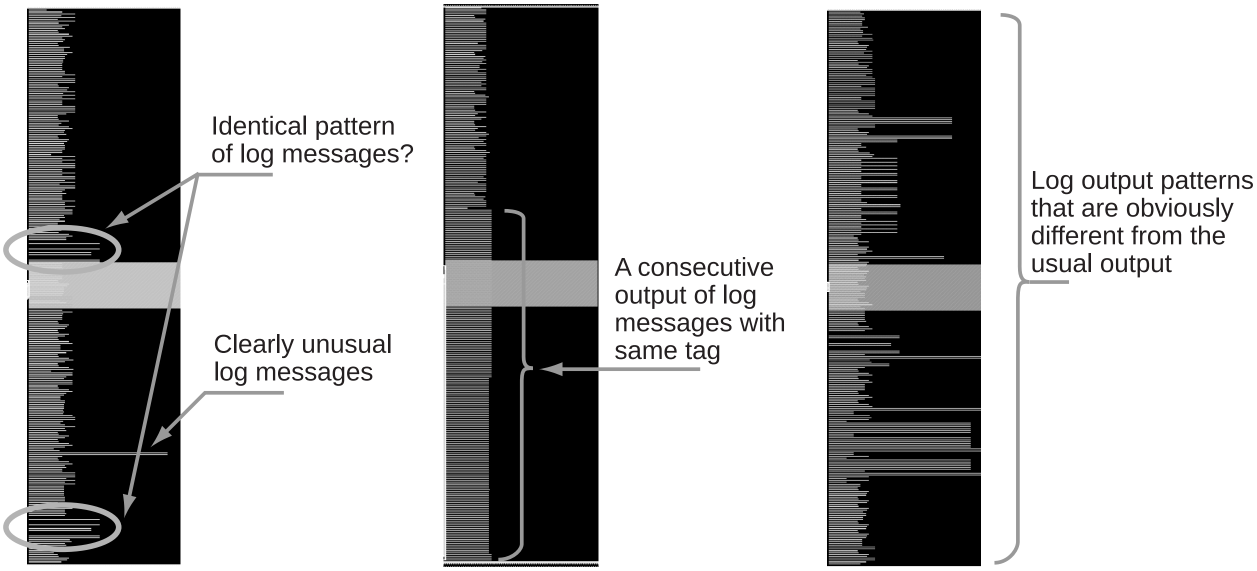


Figure 10: An example of an investigation focusing on log message outlines.

3. There is a message that is clearly longer than the others towards the bottom of the screen.

An administrator should suspect the latter two of the three as an indication of unusual log messages and inspect them in detail. To look at the log messages in detail, the inspector clicks the line in the outline area with the mouse and the text will appear in the message area. if the inspector decides there is something unusual about the messages, he or she needs to investigate the log from the following points of view:

- Were a series of messages being recorded at regular intervals?
- At what time do their messages begin to record?
- Are there any other unusual messages around the time when it started recording their messages?

It is also easy to answer the above questions using interactive functions. The inspector can extract the series of messages using word filtering. The inspector can then easily get the output trend and periodicity of such messages from the time area. He or she, of course, can easily access each message pattern and its surrounding messages. These functions help the inspector to look for unusual message around that time period.

Next, we look at the center image in Figure 10. This figure is clearly different from the log messages outlined in the other two visualizations. The inspector, therefore, easily recognizes an abnormal status at a glance. The reason is that the log messages in this outline example have various lengths of lines with red color. The inspector knows that these particular log messages rarely appear in this log-file. The inspectors should investigate them in further detail.

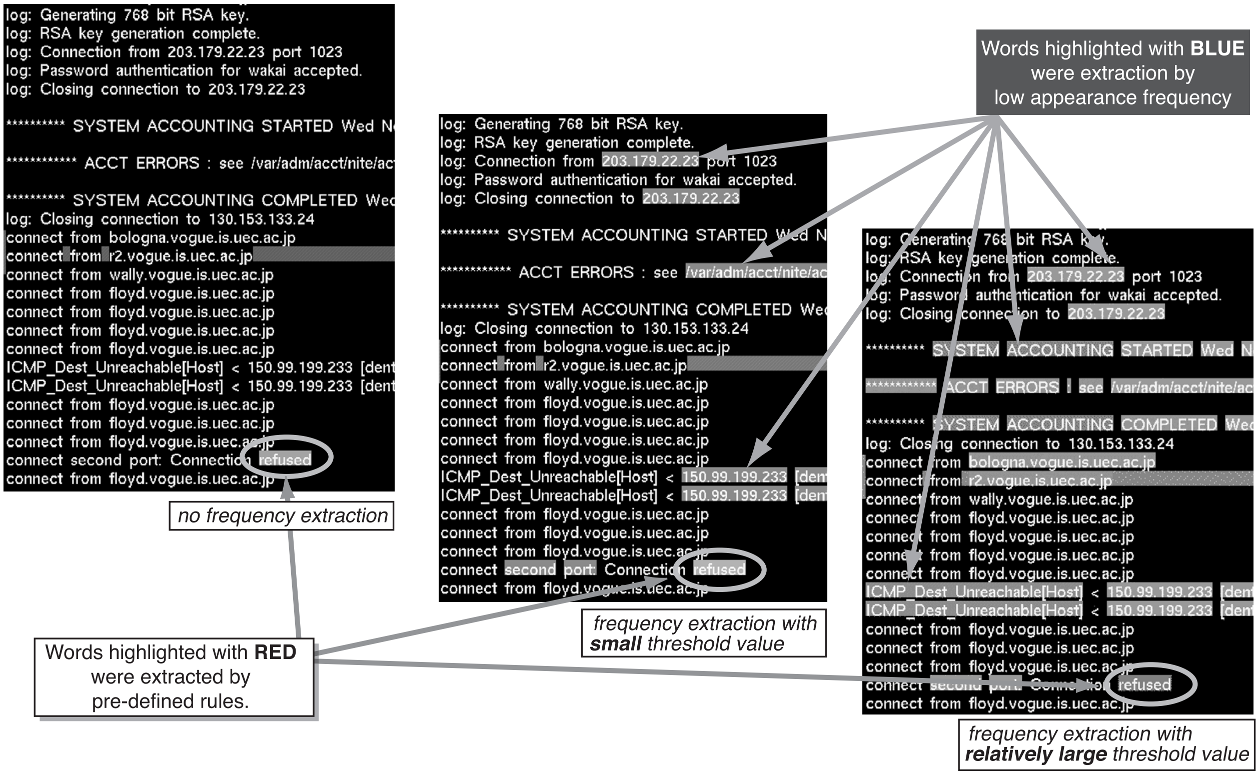
We finally focus on the right image in Figure 10. This visualization contains many lines with the same blue color and the same length towards the bottom. This is absolutely unusual status. The line colors are blue and therefore, appear to be a normal status.

We think, however, that the reason why the line colors are blue is because the repeated output of the same message makes its appearance frequency high. The inspector should investigate these messages in further detail.

As seen in these examples, outline visualization enables the inspector to recognize the log messages as a pattern. This feature provides indications to find an unusual message before reading the textual messages. In other words, outline visualization provides another method for extracting unusual log messages other than the frequency information data. The above examples give a glimpse of such ability. This capability greatly depends on using information visualization and introducing a human decision into the judgment.

An Inspection Example Using Log Message Representation with Word and Phrase Highlighting

Finally, we focus on the visual representation of the message area. MieLog represents log messages as text.



```
log: Generating 768 bit RSA key.
log: RSA key generation complete.
log: Connection from 203.179.22.23 port 1023
log: Password authentication for wakai accepted.
log: Closing connection to 203.179.22.23

********** SYSTEM ACCOUNTING STARTED Wed N

************ ACCT ERRORS : see /var/adm/acct/nite/ac

********** SYSTEM ACCOUNTING COMPLETED Wec
log: Closing connection to 130.153.133.24
connect from bologna.vogue.is.uec.ac.jp
connect from r2.vogue.is.uec.ac.jp
connect from wally.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
ICMP_Dest_Unreachable[Host] < 150.99.199.233 [dent
ICMP_Dest_Unreachable[Host] < 150.99.199.233 [dent
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect second port: Connection refused
connect from floyd.vogue.is.uec.ac.jp
```

```
log: Generating 768 bit RSA key.
log: RSA key generation complete.
log: Connection from 203.179.22.23 port 1023
log: Password authentication for wakai accepted.
log: Closing connection to 203.179.22.23

********** SYSTEM ACCOUNTING STARTED Wed N

************ ACCT ERRORS : see /var/adm/acct/nite/ac

********** SYSTEM ACCOUNTING COMPLETED Wec
log: Closing connection to 130.153.133.24
connect from bologna.vogue.is.uec.ac.jp
connect from r2.vogue.is.uec.ac.jp
connect from wally.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
ICMP_Dest_Unreachable[Host] < 150.99.199.233 [dent
ICMP_Dest_Unreachable[Host] < 150.99.199.233 [dent
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect second port: Connection refused
connect from floyd.vogue.is.uec.ac.jp
```

```
log: Generating 768 bit RSA key.
log: RSA key generation complete.
log: Connection from 203.179.22.23 port 1023
log: Password authentication for wakai accepted.
log: Closing connection to 203.179.22.23

********** SYSTEM ACCOUNTING STARTED Wed N

************ ACCT ERRORS : see /var/adm/acct/nite/ac

********** SYSTEM ACCOUNTING COMPLETED Wec
log: Closing connection to 130.153.133.24
connect from bologna.vogue.is.uec.ac.jp
connect from r2.vogue.is.uec.ac.jp
connect from wally.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
ICMP_Dest_Unreachable[Host] < 150.99.199.233 [dent
ICMP_Dest_Unreachable[Host] < 150.99.199.233 [dent
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect from floyd.vogue.is.uec.ac.jp
connect second port: Connection refused
connect from floyd.vogue.is.uec.ac.jp
```

Words highlighted with **BLUE** were extraction by low appearance frequency
no frequency extraction
frequency extraction with **small** threshold value
Words highlighted with **RED** were extracted by pre-defined rules.
frequency extraction with **relatively large** threshold value

Figure 11: An example of an investigation focusing on log messages.

However, more than just text, MieLog uses highlighting features, which makes it possible to quickly recognize words and phrases with low appearance frequency.

Figure 11 gives three examples of visual representations of the message area. The difference between these visualizations is the method of highlighting the words and phrases for indicating the suspicious log messages. The left visualization is an example using keyword extraction only. There is one word highlighted in red. It means that the word is one of the keywords pre-defined by the inspector.

The other two visualizations are examples using keyword and frequency extraction. There are some words colored blue. It means that they are low appearance frequency words. In the center visualization of Figure 11, the inspector sets the threshold value for low frequency extraction to low. In the right visualization, the inspector sets the threshold value to middle. There are, of course, many more highlighted words with blue color in the right visualization than the center.

If the inspector wants to know the low frequency word, he or she must define the threshold value first. In the current implementation, the inspector must select the threshold value from fixed values using pop-up menu. Then, the inspector clearly recognizes the words and the phrases with low appearance frequency because they are highlighted in blue.

We believe that indicating low appearance frequency words and phrases helps administrators find unusual log messages. Such words and phrases are probably related to unusual log messages. Visual representations reduce the chance of overlooking them in manual log inspections and makes it easier to recognize low frequency words and phrases where they appear.

Discussion

We describe the advantages and proposed future work of MieLog in this section.

Advantages of MieLog

MieLog provides administrators with various tools for inspection using information visualization and statistical analysis. They help administrators to look for unusual log messages. MieLog also makes it possible to inspect logs interactively. The advantages of MieLog are as follows:

- **Logs in various formats simultaneously.**. The data analyzed in MieLog consists of logs that were converted into an intermediate format. If there is a log we want to inspect with MieLog, we must first convert it. We can inspect any log using MieLog if the conversion process is provided. It also enables the inspector to inspect more than one log at a time. In other words, we can integrate multiple logs into one, based on the recorded time. It reduces the time and operations involved in inspecting multiple logs. Moreover, such log integration shows the relationship between the messages that were recorded in separate log-files.
- **Methods which assist in finding unusual log messages.** MieLog extracts appearance frequency information from the log in various graphic visualizations. Their information makes it possible to provide various indications of abnormal events which may be almost impossible to find by administrators reading the textual records. They also allow the inspection of logs from a global point of view. The inspector, for example, can decide whether the log message is unusual or not base on the number of log messages from a specific program. No prior operations and knowledge are needed in this process. Even if the inspector has no knowledge and experiences, he or she makes use of this advantage.
- **Visual Representation and Interactive Functions.** Even if abnormal indications are extracted through statistical analysis, it does not make sense that the inspectors can not recognize them. To resolve this problem, MieLog represents their indications visually in order to recognize them easily and quickly.
  Using information visualization makes it possible to interact with the visualized information directly. Using interactive functions, the inspector can easily and intuitively extract the log messages that fit a specific condition. We think that it helps to bring the human decision making process to the log inspection task.

We put above advantages in another way.

The greatest advantage of MieLog is to provide a method of anomaly detection for manual log inspection task. An anomaly detection is well-known technique for detecting intrusive behavior in intrusion detection research. There is, however, no system that makes use of such technique. Other log inspection systems make use of misuse detection only, such as keyword search. Information visualization and statistical analysis make it possible to use such technique in inspecting computer log manually. From this point of view, MieLog varies greatly from other log inspection tools.

One more another advantage of MieLog is that it is a human-centered system. MieLog is just a log browser, not an automated log inspection tool. And information visualization makes easier to recognize the log content than textual representation. We consider that there are still a lot of system administrators who want to inspect the log by themselves. There is, however, no system that has functions in order to meet their requests. We believe that MieLog is a tool that has a variety of functions to meet their requests.

Future Works

We describe proposed future enhancements of MieLog in this section. There are two main areas for improvement.

First issue is the log message extraction method. We use the appearance frequency information for extracting unusual log messages. However, not all extracted log messages are unusual log messages. We must evaluate the validation between extracted messages and unusual log messages and refine the extraction method. For example, many numerical values in the log are extracted as low frequency words because they are not recorded repeatedly. Therefore, we should exclude them as targets of statistical analysis.

The next issue is the performance related to the size of logs. The modules that are mainly affected by the size of log are statistical analysis and interactive functions. As the size of log increases, the more time is needed to process the response of the interactive functions. To alleviate this problem, we think it would be better to separate the statistical analysis from the graphical browser. We are also considering using a database.

The last issue is to add a real-time log monitoring feature. This issue has a lot of problems. We think that we must modify the system design largely in order to implement this feature. We also consider that we should be prepare for the another visual expression method for real-time log monitoring. One reason is that MieLog visualizes only a small number of log messages in current representation method. If we use MieLog in real-time log monitoring, MieLog should have an ability to visualize a large number of log messages more than the current because it is easily expected that a lot of log messages are suddenly outputted at a time. If such case occurred, administrators lost the chance to see unusual log messages like above.

In addition, we must prepare log conversion programs for the various type of logs in order to capitalize on the advantages of MieLog. We currently provide only log conversion for UNIX syslog formats.

Related Works

There are a number of log inspection tools already in existence which can be compared with MieLog. In this section, we describe two typical log browsing and inspection systems, and explain how MieLog differs from these types of systems.

One system is "Xlogmaster" [9]. This is a GUI based log monitoring system running on X window system. The main problem of Xlogmaster is that it represents log messages as text and it is harder for inspectors to recognize the log messages. Moreover, it is almost impossible to determine characteristics in the log, such as a message output trend and so on. The inspector must define the keywords in order to extract the unusual log messages. Administrators who have no knowledge and experience of performing log inspection will have difficulty extracting unusual log messages.

The other system is "SeeLog" [1]. SeeLog, like MieLog, represents log messages visually. However, the main problem with SeeLog is that only an outline visualization is available. It is thus difficult to ascertain characteristics of the log. Although SeeLog enables a visual representation of textual log messages, it is difficult for administrators to browse through them. It also has the problem of requiring keyword definitions in order to extract unusual log messages.

There are other log monitoring tools such as Swatch [5], Logsurfer [7] and syslog-ng [8]. However, a novice administrator will have difficulty using these tools effectively because the method of unusual log message extraction in these tools is by keyword search.

There are a number of problems with using keyword search as the inherent method of extraction. The first problem is that the inspector must define the keywords. It is, however, difficult for some administrators to do this because not all administrators are aware of the keywords for unusual log message. The second problem is that it is almost impossible to extract unusual log message that are not widely known. The third problem is that extracted log messages are represented as text and will still have a problem with the recognition load of the log messages. The last problem is that these systems do not support log inspection of messages around the suspect message. Administrators must inspect log messages manually around the time of the recorded extracted log message in order to find other related unusual log messages. They also might have to look for another log-files in the same purpose.

Conclusion

In this paper, we have described the interactive visual log browser, named MieLog. MieLog assists human inspection of computer log data. MieLog has three main features which address some problems of log inspection tasks:

1. MieLog reduces the recognition load of the log by using information visualization.
2. MieLog's General Log Format allows the administrator to inspect various kinds of logs at one time.
3. MieLog uses statistical analysis to extract various indications that might closely relate to unusual log messages.

These features provide the following merits. The inspector can inspect more than one log at a time. It is also possible to find an unusual log message even if the inspector has no prior knowledge about them. The most important merit of MieLog is that it brings the human decision making process into the log inspection task.

Future works on MieLog include its evaluation in a practical environment, and the refinement and extension of message extraction methods and interactive functions.

Availability and Requirements

Regrettably, MieLog is not freely available because we have a plan to be a commercial product. However, we might release a limited version of MieLog in future. The reason is that we have to evaluate it and collect the opinions about MieLog.

Please feel free to contact the author by E-mail to zetaka@computer.org for the current status of MieLog or any related information.

Author Information

Tetsuji Takada is a researcher at the Satellite Venture Business Laboratory in University of Electro-Communications, Tokyo, Japan. His main interests are information visualization and computer security. He received his BS, MS and Dr. of Eng. degrees in information engineering from the University of Electro-Communications in 2000. He is a member of the IEEE. Tetsuji can be reached via email at zetaka@computer.org .

Hideki Koike is an associate professor at the Graduate School of Information Systems, University of Electro-Communications, Tokyo, Japan. His research interests include information visualization, vision-based human-computer interaction, and computer security. He received his BS degree in mechanical engineering from the University of Tokyo in 1986. He received his MS and Dr. of Eng. degrees in information engineering from the University of Tokyo in 1988 and 1991, respectively. He is a member of the IEEE and ACM. Hideki can be reached via email at koike@acm.org .

References

[1] Eick S. G., M. C. Nelson, and Schmidt J. D.:, "Graphical Analysis of Computer Log Files," *Communications of the ACM*, Vol. 37, No. 12, pp. 50-56, 1994.

[2] Lee, W. and S. Stolfo: "Data Mining Approaches for Intrusion Detection," In *Proceedings of Seventh USENIX Security Symposium*, pp. 79-93, Jan 1998.

[3] Cox, Kenneth C., Stephen G. Eick, Graham J. Wills, and Ronald J. Brachman, "Visual Data Mining: Recognizing Telephone Calling Fraud," *Journal of Data Mining and Knowledge Discovery*, Vol. 1, No. 2, pp. 225-231, 1997.

[4] James, A. H., *Audit Log Analysis Using the Visual Audit Browser Toolkit*, Computer Science Department U. C. Davis Technical Report (CSE-95-11), 1995.

[5] Hansen, S. E. and E. T. Atkins, "Automated System Monitoring and Notification With Swatch," *USENIX Seventh System Administration Conference*, Nov 1993.

[6] Toshiyuki, Masui, "LensBar – Visualization for Browsing and Filtering Large Lists of Data," In *Proceedings of InfoVis 1998*, pp. 113-120, Oct 1998.

[7] Ley, W. and U.Ellerman, *Logsurfer*, http://www.cert.dfn.de/eng/logsurf/ , 1995.

[8] "BalaBit: The Free Software Company," *syslog-ng,* http://www.balabit.hu/products/syslog-ng/ .

[9] Greve, Georg C. F., *The Xlogmaster*, http://www.gnu.org/software/xlogmaster/ , 1998.