Implementation of Event Monitoring System using Apache Atorm

Sung-Jun Kim[1], **Jae-Kook Lee**[1], **and Tae-Young Hong**[1]
[1]Supercomputing Center, Korea Institute of Science and technology Information, DaeJeon, Rep. of KOERA

Abstract - *In KISTI Supercomputing center, our system administrators are monitoring various log messages to detect abnormal user-behavior and system faults. The collected logs include firewall logs, system access logs, system error messages, and so on. Based on these logs, System failure and abnormal user behavior are monitored.*

Previously, these logs were stored in a database, and periodic queries were made to detect system failures and abnormal user behavior. However, as the system becomes larger and larger, the amount of collected logs is rapidly increased, so that the performance limit of the conventional method are expected.

In this paper, we implemented the prototype of the monitoring system using Apache Storm. Through this, a real-time monitoring system is constructed to monitor system failure and abnormal user behavior of large-scale cluster system.

Keywords: supercomputer, apache storm, monitoring

1 Introduction

As the only national supercomputing center in Korea, we provide HPC systems to the researchers in industries, academia, institutes and government organizations. We also operate a variety of security solutions to provide a secure service environment. In the our control room, log messages sent from various devices are collected and fault notification and firewall blocking are performed through system failure and abnormal user behavior monitoring.

During 2017, 26,248,740 system error logs occurred on the supercomputer and 235,738 illegal access attempts from all over the worlds were detected. It is impossible for a operator to identify such massive events (system errors, system connections, firewalls, etc.) and to determine whether they are abnormal or not. In order to detect such anomalous event quickly, the system operated by storing events in the MySql database and periodically checking them.

However, it is expected that the amount of failure logs to occur on the new supercomputer (approximately 8,000 nodes) will increase dramatically, which would delay the event processing time to handle large volumes of logs with existing monitoring system.

In this paper, we have designed and implemented an abnormal event detection system based on apache storm that can handle real - time event processing to solve the expected processing time delay after introducing a next supercomputer. Using the event monitoring system implemented in this paper, it will help to operate more stable system by detecting and responding to abnormal events quickly.

2 Background

2.1 Apache Storm

Apache Storm is a open source distributed realtime computation system. Storm makes it easy to reliably process unbounded streams of data, doing for real-time processing what Hadoop did for batch processing [1][4]. Storm can used with any programming language.

2.2 Logstash

Logstash is an open source, server side data processing pipeline that ingests data from a multitude of sources simultaneously, transforms it, and then sends it to other solutions. The Logstash events processing pipeline has three stages: inputs → filters → outputs. Inputs generate events, filters modify them, and outputs ship them elsewhere [2].

2.3 Esper

Esper is an open source software product for Complex event processing and Event stream processing that analyzes series of events for deriving conclusions from them [3]. Esper extends the SQL standard for its engine and enterprise framework, providing Aggregate function, Pattern matching, event windowing and joining [3].

3 Implementation

The purpose of the proposed system is to detect system failure and to detect intrusion of our supercomputer. Failure detection monitors the contents of system log messages (/var/log/message) collected by our entire systems. This module checks for the following:

- Does the log message contain any keywords that system administrator has predefined?

- Are the monitoring metrics (CPU load, memory usage, etc.) exceeding the threshold?

Intrusion detection checks for the following conditions based on the connection log (/var/log/secure) and firewall log :

- Is the same IP trying to connect with multiple user id?
- Did user try to connect multiple times from abroad source IPs during a short time interval?
- Did the same user fail to connect more than once at multiple source IP?

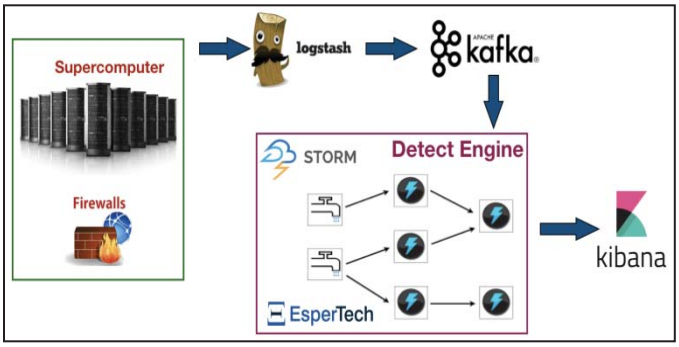

Figure 1 Architecture of proposed system

<Figure 1> shows the proposed system architecture. Logstash ingests original logs from various devices, processes them, and sends them to Kafka, which is responsible for the queuing system. We use Kafka to avoid data loss. The spout of apache storm receives log messages from Kafka topic. In the bolt, the time-windows base check is performed using Esper CEP engine, and the event are trigger when the failure condition is satisfied. When an event occurs, it can monitor in the control room through Kibana which web based visualization tool.

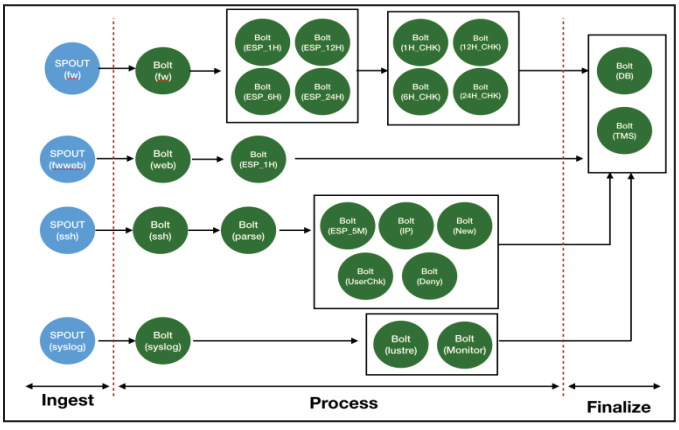

Figure 2 Apache storm topology

<Figure 2> shows apache storm topology of proposed system. It consists of 4 spout and 20 bolts. The spouts are mapping to a log to be collecting. The spouts ingest log messages from firewall, web application firewall, secure log and syslog message.

The bolts are divide into the processing steps to perform. In process step, when it is necessary to check the occurrence frequency of an event within a certain period, the threshold value of event occurrence inspects within a certain period using Esper. In finalize step, the processed event information is stored in the databases and the event occurrence notification is perform.

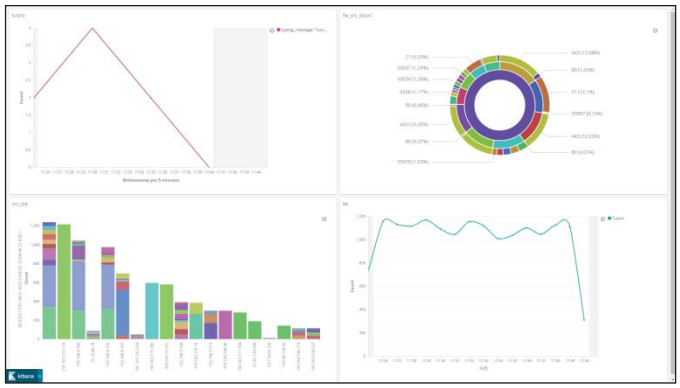

Figure 3 Kibana dashboard

<Figure 3> shows Kibana dashboard screen shot. The dashboard visualizes the events generated by the storm that analyzed the messages. The control room monitors the occurrence of events through the dashboard.

4 Conclusions

In this paper, we implemented a monitoring system using apache storm. The developed monitoring system will be use in the control room and it will help to recognize event occurrence. By doing so, we will be able to provide more secure and reliable serves to our users. In the future, we will achieve faster service stabilization by applying this system to the next supercomputer.

Acknowledgement

This research was supported by Korea Institute of Science and Technology Information (KISTI)

5 References

[1] Apache Storm, http://storm.apache.org/

[2] Logstash, http://www.elastic.co/products/logstash

[3] Esper, http://www.espertech.com/esper/

[4] Zhang, Xinyao, "The Analysis of Parallelism of Apache Storm",2016