Improving the logging infrastructure of HPC and Linux services

M. Dessalvi
GSI, Darmstadt, Germany

Introduction

Logging system events, especially for big IT infrastructures, is essential. Whenever a problem arise System Administrators turn their looks towards log files but as IT infrastructures grows in size and complexity the volume of the available logs will increase dramatically as well the resources needed to analyze them.

Overview

The GSI HPC group have already implemented multiple solutions, based on open source software, to analyze different kind of logs and events. A brief list of those software inlcude: Nagios, Netdisco, Collectd, MRTG graphs and Torrus.

Analyzing text logs files, produced by hundreds of hosts, is a daunting task. Traditionally, looking into such a massive amount of informations requires the use of specific utilities for searching, parsing, manipulating and extracting useful data. On UNIX/Linux systems these tools are usually available directly from the shell: cat, tail, grep, sed, awk , sort and many others. Unfortunately, these tools are not enough to spot trends and make correlations with different events scattered on multiple files.

To overcome this problem a solution was implemented, based on Logstash[1], Redis[2] and Elasticsearch[3], plus Kibana[4] as a web interface.

Logstash

Logstash is a tool for managing events and logs. It can be used to collect logs, parse them, and store them for later use.

The main idea behind this tool rely on the concept of plugins. A combination of different plugins let the admins create their own log pipeline and extract informations coming from different sources and store the results with different storage solutions.

A short list of the available plugins:

- Input: TCP/UDP, text files, Syslog, MS Windows Event logs, STDIN, etc.
- Filters: alter, collate, geoip, key value, metrics, multiline, XML, Zeromq, etc.
- Output: CSV, email , Graphite, StatsD, Elasticsearch, Nagios, XMPP, etc.

Different Logstash agents may be deployed to deal with logs from different sources and structures. The parsed results, in JSON format, is then pushed to Redis which acts as a broker between multiple Logstash agents and the Elasticsearch server.

Elasticsearch

Elasticsearch is basically a text indexing engine. It is based on Apache Lucene[5], a full-featured text search engine library written entirely in Java. Elasticsearch most interesting characteristic is its fast search responses, based on the concept of *inverted index*: instead of searching text strings directly, it creates an index from incoming text and performs searches on the index rather than the content. The Kibana web interface add an extremely flexible web interface for visualizing the collected data in real time.

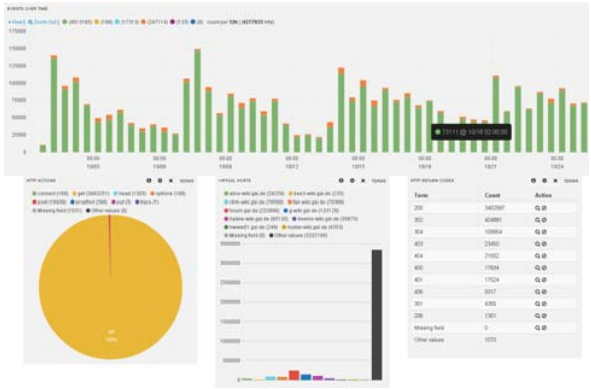

Figure 1: An example of the Kibana web interface for the Apache logs.

Outlook

Future developments of this solution will include a migration to the new 1.0 stable branch of Elasticsearch and extract even more informations from the logs through the combined use of Graphite[6] and Statsd[7].

References

[1] http://www.logstash.net
[2] http://www.redis.io
[3] http://www.elasticsearch.org/overview/elasticsearch/
[4] http://www.elasticsearch.org/overview/kibana/
[5] http://lucene.apache.org/core/
[6] http://graphite.wikidot.com/
[7] https://github.com/etsy/statsd/

 doi:10.15120/GR-2014-1-IT-02