A framework to analyze massive data from applications and services of a meteorological data center

Eugênio Sper de Almeida
Center for Weather Forecasting and Climate Studies (CPTEC)
Brazilian National Institute of Space Research (INPE)
Cachoeira Paulista, SP, Brazil
eugenio.almeida@cptec.inpe.br

Ivo Koga
Center for Weather Forecasting and Climate Studies (CPTEC)
Brazilian National Institute of Space Research (INPE)
Cachoeira Paulista, SP, Brazil
ivo.koga@cptec.inpe.br

***Abstract*—The Center for Weather Forecasting and Climate Studies (CPTEC) produces and disseminates results from Numerical Weather Prediction (NWP) models simulations to Brazil and South America. In order to accomplish this task, its data center hosts a supercomputer and many other computational resources that receives data from many heterogeneous sources. To find and solve problems, it is important to monitor several computational resources of the data center twenty-four hours a day, seven days a week. This paper presents a framework to collect and analyze massive data from a meteorological data center. The results show the analysis of NetFlow and File Transfer Protocol (FTP) server log data, which helped the CPTEC monitoring team advance their knowledge of infrastructure usage and problems solution.**

Keywords—meteorological data center; Elastic Stack; logs; NetFlow, data analytics.

I. Introduction

Accuracy has improved in weather forecasting with the development of Numerical Weather Prediction (NWP) models. They represent the movement and the physical processes of the atmosphere through mathematical equations and computer software [1]. The NWP models simulation generates information related to the evolution of the atmosphere and ocean phenomena, which can conduct to more precise weather forecast predictions.

A complex deadline-sensitive meteorological workflow produces the numerical predictions [2]. It involves different applications: the meteorological data acquisition by a pre-processing system, the execution of NWP models and the post-processing of the outputs.

The World and Regional Meteorological Centres[1] execute NWP models in global and local scale, which have a considerable computational cost and complexity. Additionally, they execute real-time and non real-time functions of the Global Data Processing and Forecasting System (GDPFS) [3].

The meteorological workflow usually executes four times a day (00, 06, 12 and 18 hours UTC). Its timely execution depends on the supercomputer and considerable amount of computational resources (computing systems, networks and storage devices) hosted in the meteorological data center [4].

The execution of the meteorological workflow may be compromised if something wrong occurs. This is why operations and management teams of meteorological data centers have to perform their duties, including monitoring of computing resources to identify potential sources of problems.

The monitoring task is generally conducted using different monitoring tools to ensure an uninterrupted operation. Additionally, it is important to analyze and understand the information provided by the monitoring tools [5].

These monitoring tools usually use system logs. One example is the detection of high memory usage or swapping activity in a given computational resource, which can indicate some anomaly on its behavior.

Since logs are produced by a large number of equipments, the result is a huge amount of data to be analyzed. They are also produced at a very high rate, e.g. one line of log per millisecond. This poses a challenge for log collection, processing and storage [6].

Many traditional monitoring systems (Nagios[2], Cacti[3], Ganglia[4], Munin[5] and Zabbix[6]) collect and provide computational resources health monitoring. They assist system administrators in assessing and improving the management of their infrastructure [7].

The operations and management teams use different monitoring tools to perform their roles. Besides notifying the status of services provided, they also need to analyze and understand information related to processing, network, usage, account management, and problem detection and resolution [5].

Although they provide interesting metrics about the usage of the datacenter, it is still difficult to mitigate problems. Obtaining a complete overview of events in the data center by correlating different log data is not an easy task.

[1] http://www.wmo.int/pages/prog/www/DPS/gdps-2.html

[2] https://www.nagios.org

[3] http://www.cacti.net

[4] http://ganglia.sourceforge.net

[5] http://munin-monitoring.org

[6] http://www.zabbix.com

DOI 10.14684/WCSEIT.4.2017.20-25

© 2017 WCSEIT

Some tools provide information related to network monitoring, while others provide information from servers. They do not usually provide tools to jointly monitor all the data center resources. An integrated approach is required to analyze the information and develop correlations about the events in order to have insights about problems.

In this paper, we present a framework that collects, integrates and analyzes NetFlow and File Transfer Protocol (FTP) server log data from the data center of the Center for Weather Forecast and Climate Research (CPTEC), the major Brazilian NWP generator.

In the next section we present some related work, including platforms for indexing and search. Section III presents datasets information, detailing network flows and FTP log data. Section IV presents the architecture we have implemented to gather, analyze and publish information from FTP server log and NetFlow data. In section V, we present the results and discuss some interesting insights from the CPTEC data center network traffic. We conclude this paper in section VI.

II. Related Work

System logs are one of most valuable sources of information in a data center. We can understand what is happening in the data center, including system failures and their causes, gathering such information.

Each meteorological data center has their own solution to collect, process and analyze their infrastructure's health. Some acquire information using traditional monitoring tools and access this information when necessary. Others integrate monitoring data using relational or NoSQL databases, and index them using indexing and searching systems.

To perform the index and search capabilities, Apache Lucene [8] provides a text search engine for search and query. Apache Solr [9] and Elasticsearch [10] extend Lucene functionalities, including additional features: scalability, user interface (UI), administration and filtering. Elasticsearch has created the Elastic Stack platform (Logstash, Elasticsearch and Kibana), which provides different tools to deal with logs.

The infrastructure of Global Science experimental Data hub Center (GSDC) at the Korea Institute of Science and Technology Information (KISTI) used the Elastic Stack to query and visualize service status, and Splunk[7] and Observium[8] to monitor storage and network, respectively [11].

Gudivada et al. [12] used the Elastic Stack to parse Web access logs and to infer the repeatable user behavior, transforming them into Customer Behavior Model Graphs (CBMG).

Sanchez et al. [13] presented a flexible and scalable monitoring infrastructure for the HPC computational platforms at Los Alamos National Laboratory (LANL). Taerat et al. [14] developed a log message pattern extraction of HPC systems logs.

Following the same trend of these works, we are carrying out an initiative to integrate all information about our data center into an Elastic Stack based infrastructure

III. Log data preparation

We used the Elastic Stack to analyze the datasets (Fig. 1). We filled the first dataset with logs from the FTP server, generated locally on the external demilitarized zone (DMZ). We had to export data from the DMZ to the internal network and then consume it through Logstash.

We inserted NetFlow data in the second dataset, streamed direct from the main Core Switch to Logstash, acting as a Netflow log collector. NetFlow collects traffic data generated by computational resources, applications and services.

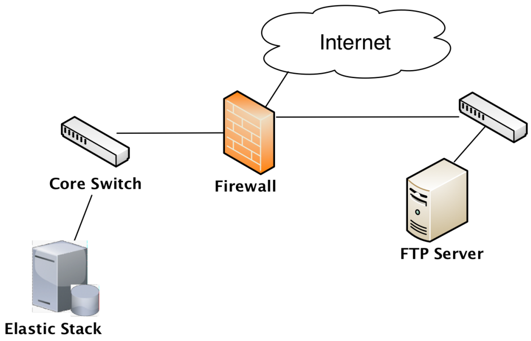

Fig. 1. Location of datasets sources and Elastic Stack

The FTP server uses the ProFTPD (short for Pro FTP daemon), a free and open-source software, that generates the transfer log file (xferlog) in ASCII format, which contains logging information.

The information about service usage includes: current local time, total time in seconds for the transfer, remote host name, size of transferred file in bytes, name of the transferred file, transfer type (a = ascii; b = binary), special action flag: (C = compressed; U = uncompressed; T = tar'ed; = no action was taken), direction of the transfer (o = outgoing; i = incoming; d = deleted), method by which the user is logged in (a = anonymous; r = real), username, service name (usually ftp), authentication method (0 = none; 1 = RFC931 Authentication Authenticated-user-id: user id or ‘*’), completion status (c = complete; i = incomplete).

For the first dataset, we programed Logstash to execute the following tasks: read the xferlog files (FTP logs), convert to numbers the transfer time and file size, and generate timestamps of date/time in ISO 8601 format.

The second dataset consists of NetFlow data in version 5 format: number of flows exported in this packet (count), source IP address (srcaddr), destination IP address (dstaddr), TCP/UDP source port number (srcport), TCP/UDP destination port number (dstport), packets in the flow (dPkts) and IP (protocol) type.

Cisco developed the Netflow network protocol, which provides means to examine aggregated IP traffic, forwarded

[7] http://www.splunk.com

[8] http://www.observium.org

within NetFlow-enabled routers and switches, and aggregate them in network flows [15].

NetFlow allows IT professionals to understand the network traffic and the amount of data exchange inside the network. Unlike packet capture or deep packet inspection (DPI), collecting network flows ensures privacy since sensitive data is not collected [16].

We set up Logstash using NetFlow codec to decode Netflow v5/v9/v10 (IPFIX) flows and GeoIP to locate a host geographic location from its IP address. The timestamp was also generated in ISO 8601 format.

In both cases Logstash forwarded data to Elasticsearch for storage and indexing. Using Kibana, we select the dataset and refine a data subset for analysis, presenting the results in form of graphics. We grouped all graphics in a Dashboard.

Using the first dataset, we created a pie chart representing the relationship between the log contents. Additionally, we generated bar charts of ftp use count, transfer type, direction and hosts split by IP address.

For the second dataset, we generated a pie chart that presents the relationship between the IP protocol, source IP address/port number and destination IP address/port number. We also created bar charts of flows count, number of bytes and number of packets in the flow.

The Fig. 2 presents the mechanisms to collect data and the Elastic Stack components that deal with the datasets. The data input to Elastic Stack occurs by capturing NetFlows streams transmitted by the Core Switch (1a) and reading the log files from the FTP server (1b). The Logstash captures data and convert some fields. Next, Elasticsearch stores and index data sent by Logstash (2). Finally, users query for data (3) and visualize results (4) in different interactive interfaces using Kibana.

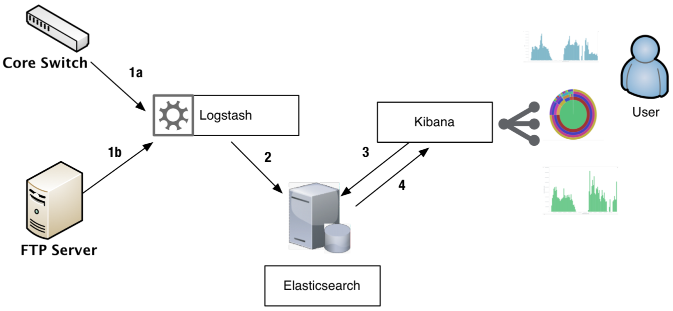

Fig. 2. The Elastic Stack Architecture

IV. Architecture for Data Center Analysis

Our initial infrastructure for data integration and analysis consisted of a Dual Quad-core server @2.27GHz with 4GB of memory and 5.4TB of storage capacity. It aimed at the analysis of file transfer data logs previously collected from the FTP server [17].

This preliminary study used Elastic Stack for parsing (Logstash 2.3.2), indexing (Elasticsearch 2.2.1) and analysis (Kibana 4.4.2). It provided some knowledge about the usage of the FTP server and data access.

In order to advance our knowledge about the data center, we conducted an experiment to understand our data center internal network traffic using on the same infrastructure. First we have configured the main Core Switch to collect network traffic and export data in NetFlow version 5 format. We identified problems related to the lack of memory, index organization and low disk space using the initial setup of Elasticsearch.

We improved our infrastructure to properly handle our data needs. We dedicated the original server to Logstash (version 3.2) and included a cluster of three Elasticsearch servers with Dual Quad-core server @2.27 GHz and 32 GB of memory each. A new version of Elasticsearch (5.2) and Kibana (5.2) were also installed. This improvement allowed the horizontal scalability and higher availability of the Elasticsearch.

Within this infrastructure, we could distribute data using a dynamic load balancing strategy available in Elasticsearch, which increased performance and availability. However, the high amount of NetFlow data produced by the Core Switch quickly exhausted the local storage capacity of the Elasticsearch servers.

We realized that the amount of storage was not sufficient. After losing some data, we decided to use the NetApp storage (100 TB) and we added 2 TB of virtual storage to each server. The current architecture (Fig. 3) provides mechanisms to store NetFlow and file transfer log data from the CPTEC data center, implemented with the NetApp storage.

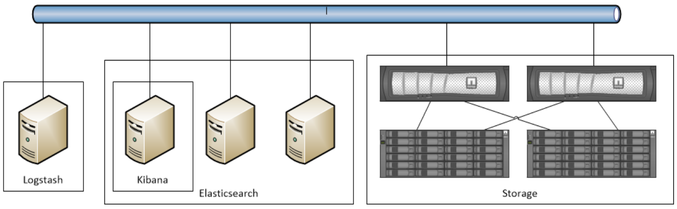

Fig. 3. Current architecture.

V. Results and Discussions

We have stored approximately 141 million of records of FTP logs from 28 February 2017 to 13 June 2017. We verified that 95.6% of accesses come from authenticated and 4.4% from anonymous users.

We certified that anonymous users are only allowed to download files. The real (authenticated) users access is divided in 71.95% of upload (incoming), 1.1% of download (outgoing) and 26.95% of removals (deleted). We observed a peak of 165,060 accesses on 5 June 2017 around 11:00 am (Fig. 4).

We split the graphic by host access and IP address, which reveals that two IP address dominates the usage of the FTP server with an exchange of about 561,844 files. We conducted a detailed analysis using the data table feature of Kibana, for a detailed view of the graphic in tabular format. We have observed a repeatedly transfer between them, indicating a mistake on the usage of the file transfer service.

The analysis with data from 28 February 2017 to 13 June 2017 period presented issues related to transferring the same data many times from the same user, storage of useless files leading to an ever increasing waste of storage size and predominance of real (authenticated) users access over anonymous users.

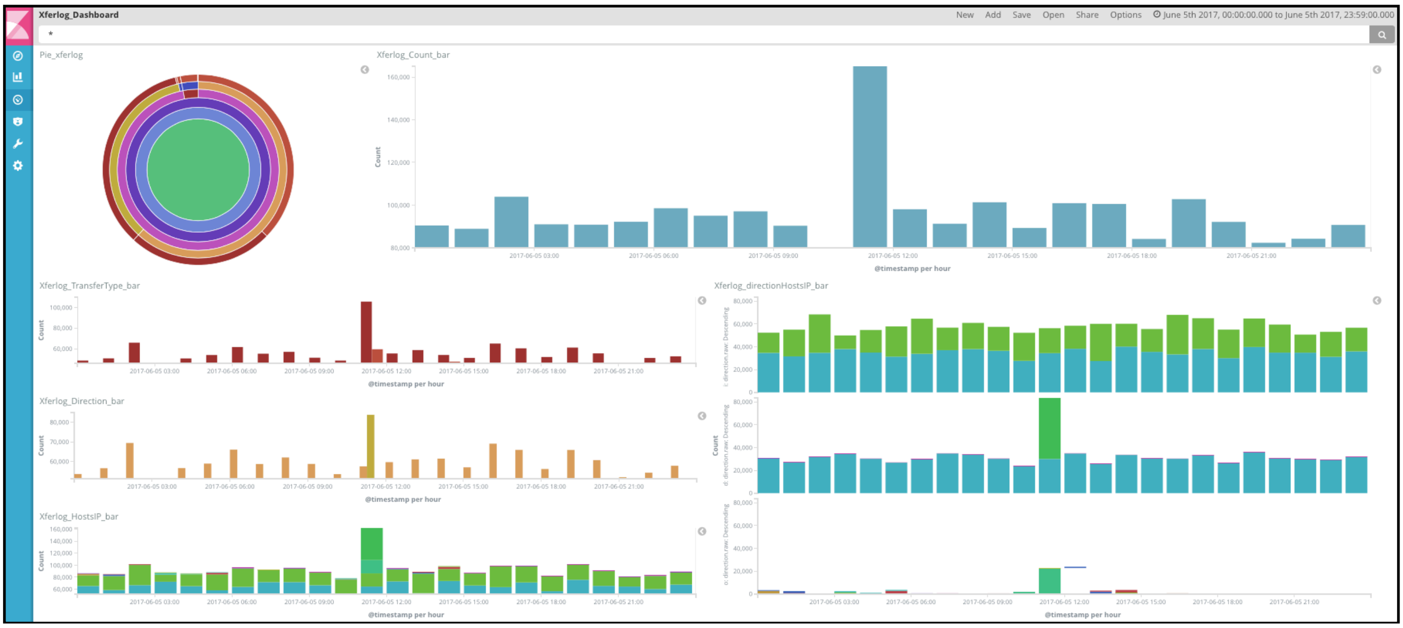

Fig. 4. Dashboard with FTP log visualizations from 5 June 2017.

Our NetFlow data analysis was conducted from 13 March 2017 to 13 June 2017. In May we verified the highest amount of data crossing the Core Switch (72 TB).

During this period, the Core Switch generated approximately 2.2 billion of network flows. This high amount of data was one of the reasons that demanded the increase of storage capacity in our infrastructure.

We presented the visualizations of NetFlow from 16 April 2017 in Fig. 5. The pie chart presents the relationship between the following NetFlow fields: transport protocol (inner circle), source IP address/port (second and third circle) and destination IP address/port (fourth and fifth circle). We have omitted IP address and ports due to privacy and security reasons.

The three bar charts correspond to count of flows, sum of NetFlow packets and sum of NetFlow bytes. The highest number of flows (40,602,491) was generated on 16 April 2017 (Figure 4), mainly from 20,508,042 TCP flows (50.51% of total) and 20,027,572 UDP flows (49.33% of total). It refers to a high amount of noncontiguous traffic produced during this day by 1,325,890,321 packets, which corresponds to 1.11 TB of data traffic.

The noncontiguous traffic on 16 April 2017 means that many connections are opened and closed during the day, which generate high number of traffic flows with small number of packets transmitted.

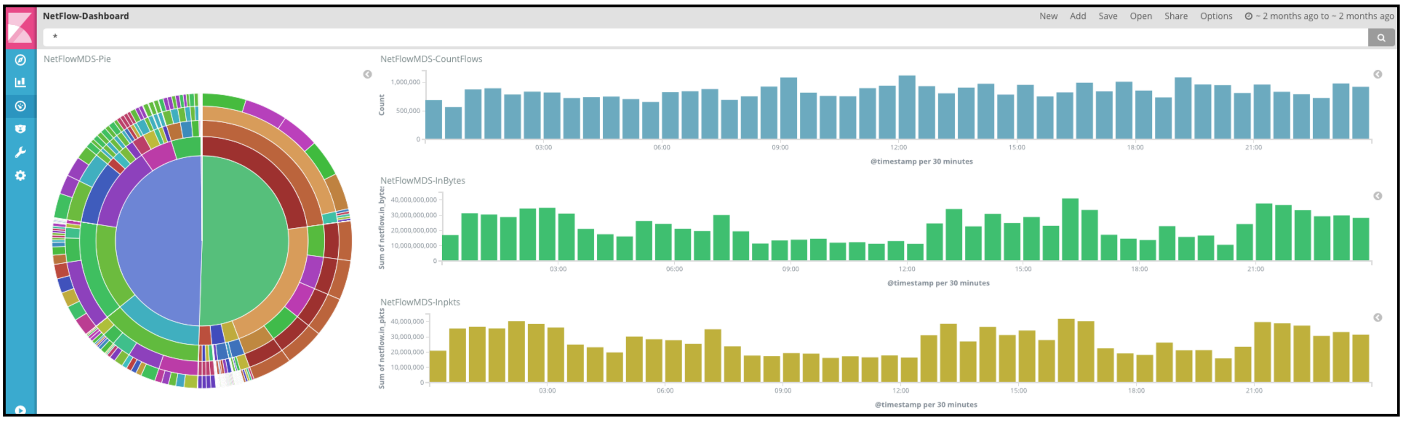

Fig. 5. Dashboard with Netflow visualizations from 16 April 2017.

We observed the highest number of packets (6,235,330,459) on 27 May 2017 (Fig. 6), which was produced by more than 7.2 TB of data crossing the Core Switch. In this case, we observed that more data is encapsulated on less flows (29,712,986).

The traffic generated by TCP applications was 7.2 TB bytes (99.87%), which was much higher than the 9.5 GB (0.13%) of the UDP traffic. In this case, the number of packets transmitted in this date was almost 5 times higher than on 16 April 2017 and refers to a contiguous traffic of the TCP protocol.

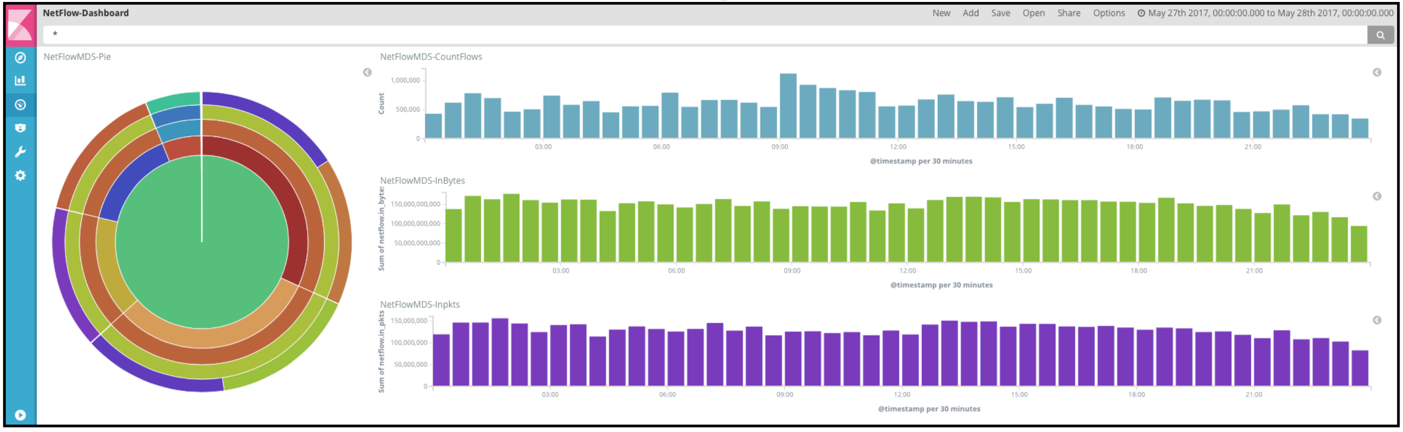

Fig. 6. Dashboard with Netflow visualizations from 27 May 2017.

In the two scenarios we could analyze data from non real and near-real time by configuring Kibana in order to present and analyze data properly. This customization reflects on the Dashboard to show events of interest integrated.

We also integrate both source of information. NetFlow visualization provides a more general view of a service usage, while details can be obtained using the FTP logs.

Fig. 7 presents integration of NetFlow and FTP log data from 05 June 2017. The first two images present the data transfers from other ftp servers. The other three images detail the information from the main FTP server: number of access, and file transfer direction and type.

Conducting for data analysis with Elastic Stack analysis allow to expose important information to understand the data center components' behavior, identify issues, guide storage capacity plan, promote data transfer better usage and support the correct use of technologies.

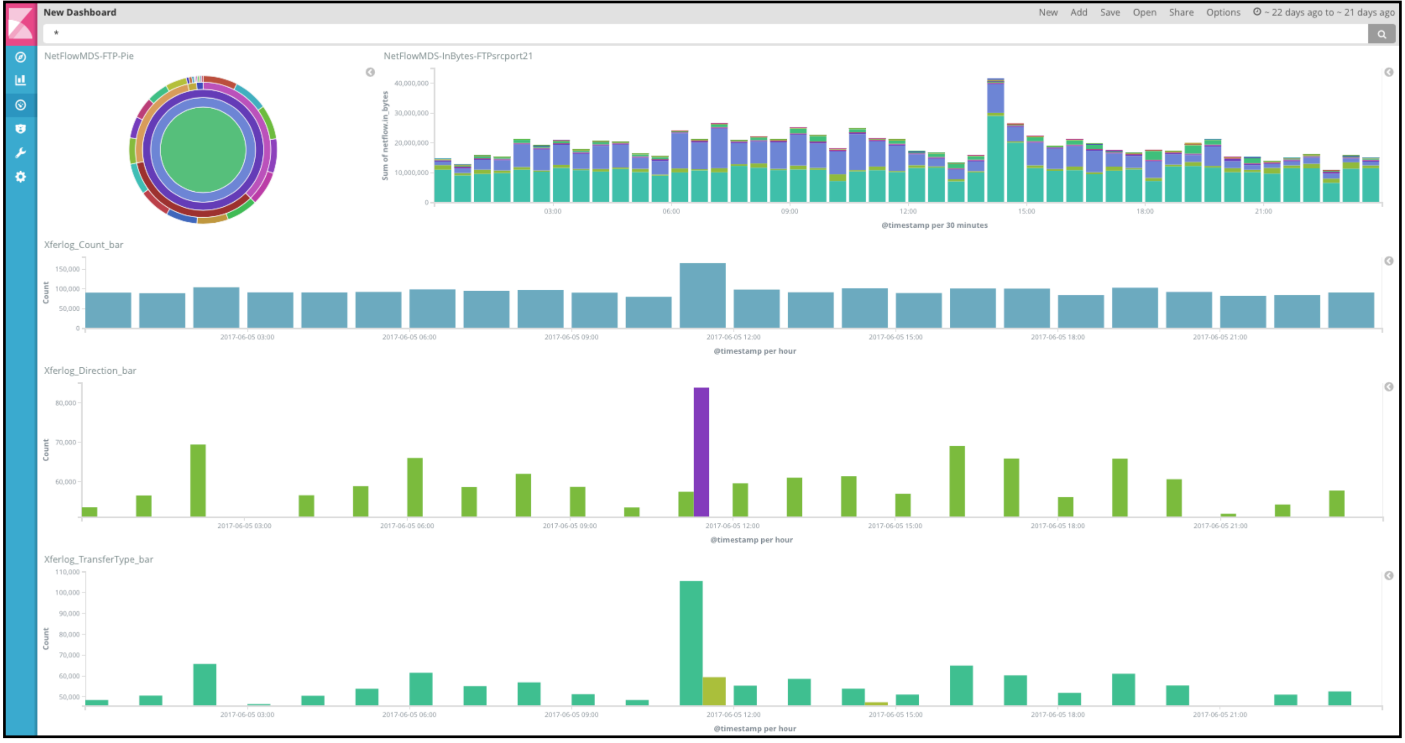

Fig. 7. Integration of Netflow and FTP logs in a Dashboard.

VI. Conclusion

In this paper we presented a framework to collect and analyze massive data from a meteorological data center. The current version allows the analysis of NetFlow and FTP server log data generated in CPTEC data center.

We configured Logstash to collect data in two different ways: reading log files (FTP log data) and capturing streamed data (NetFlow data). Logstash also transformed the data and sent to Elasticsearch for storage and indexing. Finally, we have built different graphics in Kibana, combining in a Dashboard for a complete overview.

The analysis of the FTP server log data used just one server as Elastic Stack infrastructure. However, the high amount of NetFlow data streamed from the main Core Switch required an upgrade of the original infrastructure due to the increase of data. We had to add three more servers in a cluster configuration for Elasticsearch and attach an external storage to increase storage capacity.

The analysis of FTP logs reveals issues relate to the use of the FTP server. The NetFlow analysis detailed the network traffic by analyzing the relationship between computational resources and applications/services.

Some insights related to the data center computational resources, applications and services were revealed in this work. In practice, this framework is helping the monitoring team advance their knowledge about infrastructure issues and usage. This is particularly of great importance and it is fundamental to log data to track problems and identify areas of improvement.

Acknowledgment

We would like to thank INPE/CPTEC and the support from Brazilian financing agency MCTIC/FINEP grant 0.1.16.0076.00.

References

[1] P. Lynch, "The origins of computer weather prediction and climate modeling," Journal of Computational Physics, vol. 227, n. 7, 2008, pp. 3431-3444.

[2] A. Thorpe, "Meteorology: The brainstormers." Nature 532.7597, 2016, pp. 30-31.

[3] D.P. Rogers, and V.V. Tsirkunov, Weather and climate resilience: Effective preparedness through national meteorological and hydrological services. World Bank Publications, 2013.

[4] E.S. Almeida, M.A. Bauer, and A.L. Fazenda, "Numerical weather model BRAMS evaluation on many-core architectures: a micro and macro vision," International Journal of Computational Science and Engineering, vol. 12, n. 4, 2016, pp. 330-340.

[5] C. Gough, I. Steiner and W. Saunders, "Data center management," Energy Efficient Servers. Apress, 2015, pp. 307-318.

[6] C. Vega, P. Roquero, R. Leira, I. Gonzalez, J. Aracil, Loginson: a transform and load system for very large scale log analysis in large IT infrastructures," The Journal of Supercomputing, vol. 73, 2017, pp 1-22

[7] A. Costin, "All your cluster-grids are belong to us: Monitoring the (in) security of infrastructure monitoring systems." Communications and Network Security (CNS), 2015 IEEE Conference, IEEE, 2015, pp. 550-558.

[8] M. McCandless, E. Hatcher and O. Gospodnetic. Lucene in Action: Covers Apache Lucene 3.0, Second Edition, Manning Publications Co., Greenwich, CT, USA, 2010.

[9] D. Smiley, E. Pugh, K. Parisa, and M. Mitchell, Apache Solr enterprise search server, 3 edition, Packt Publishing Ltd, 2015.

[10] H. Akdogan, Elasticsearch Indexing. Packt Publishing Ltd, 2015.

[11] S.U. Ahn, "GSDC: A Unique Data Center in Korea for HEP research." EPJ Web of Conferences. Vol. 141. EDP Sciences, 2017.

[12] V.N. Gudivada, D. Rao and V.V. Raghavan, "NoSQL systems for big data management," Services (SERVICES), 2014 IEEE World Congress on Services, Anchorage, AK, 2014, pp. 190-197.

[13] S. Sanchez, A. Bonnie, G. Van Heule, C. Robinson, A. DeConinck, K. Kelly, Q. Snead, and J. Brandt, "Design and Implementation of a Scalable HPC Monitoring System," Parallel and Distributed Processing Symposium Workshops, 2016 IEEE International, IEEE, May 2016, pp. 1721-1725

[14] N. Taerat, J. Brandt, A. Gentile, M. Wong, and C. Leangsuksun, "Baler: deterministic, lossless log message clustering tool," Computer Science-Research and Development, vol. 26, n. 3-4, 2011, pp. 285-295.

[15] Cisco Systems (2006) Introduction to Cisco IOS NetFlow - A Technical Overview. White Paper, http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html.

[16] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto and A. Pras, Flow monitoring explained: From packet capture to data analysis with NetFlow and IPFIX," IEEE Communications Surveys & Tutorials, vol. 16, n. 4, pp. 2037-2064, 2014.

[17] I. Koga and E. Almeida, "File Transfer Log Analysis: A meteorological data center case study, " Journal of Computational Interdisciplinary Sciences, 2016.