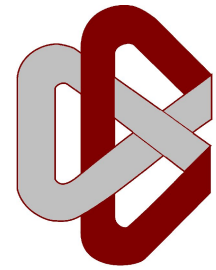


Centro de Investigación en Matemáticas, A.C.

CIMAT

Live Big Data Logs Analysis

TESIS

Que para obtener el grado de

Maestro en Ingeniería de Software

Presenta

Alfonso Alvarez Sánchez

Director(a) de Tesis

Alejandro García Fernández

Zacatecas, Zacatecas., 16 de 07 de 2015

Index

Figures and Tables Index

Abstract

AdBlock Plus (ABP), the browser extension that allows to surf the web without annoying ads, is the most popular browser extension in the world with over 350 million downloads. ABP is fanatical about privacy and data collection. Because data is knowledge, and when the collected data is bigger than what can be handled, the risk of missing important information rises, the time to process it increases, and the reaction to events takes too much time. In this research a combination of three open source tools (Logstash, Elasticsearch and Kibana), is implemented as an exploratory experiment to get as much information as possible out of the collected data providing the possibility to analyze log files in realtime. The premise for this research is to keep a solution easy to maintain, high scalable and reliable,providing the company with the information required to take better decisions.

Keywords - Big Data, Log Analysis, Real-time, Logstash, Elasticsearch, Kibana, Automatic Provisioning, Open Source

Acknowledgements

I want to acknowledge the financing provided by the Consejo Nacional de Ciencia y Tecnología (CONACyT) and from the Centro de Investigación en Matemáticas (CIMAT - Guanajuato, Mexico) for the realization of this work.

Also I want to thank MIS. Alejandro Garcia and Jose G. Hernandez, MTI for their guidance in the development of this work. To all my reviewers and supporters: Arturo, Leonel and Agus who made the writing process faster.

To Eyeo GMbH and its members. In this great company I started developing this project in company and guidance of great people. Specially Matze, who was very supportive in the creation of this work since the very beginning and motivated me to consider a research in this area. Kirill, who helped shaping the idea. Paco who helped in the development and growth of the project. And of course Arthur and Peter who provided the music to motivate. They still do.

To my family. My parents (Alfonso y Lourdes), my sister (Lulu) for providing the initial motivation for studying the Master in Software Engineering. They all still don’t know what I do for living.

Chapter 1. Introduction

Three mega tendencies are changing the technological landscape: (Zhang, Cheng and Boutaba, 2010)

- The first one “Cloud Computing” with it’s children:
    - Infrastructure as a Service (IaaS) ex. AWS, with an estimate of at least 454,400 servers in seven data center hubs around the globe. and Google Compute Engine with around 900,000 (“StorageServers”, 2013)
    - Platform as a Service (PaaS) ex. Heroku, Elastic Beanstalk,
    - Software as a Service (SaaS) ex. Sales Force.

- The second web applications, a natural consequence of the evolution of the Web: Facebook, processing around 750 TB and Netflix storing up to 100 terabyte. And
- The Third: Big Data that is used to analyze and understand the large quantities of data that the cloud and web apps are able to collect due to their centralized nature. (Duarte, 2014)

AdBlock Plus is the most downloaded browser extension in the world with over 350 million downloads, 60 million active user, over 60 servers and growing. The plugin blocks intrusive publicity in websites, like banners with movement or ads that don’t allow to see the content of the website unless you click in them. (Adblockplus.org, 2015)

The question that motivates this research is: How do these companies (ABP, Google, Facebook, Amazon), know if their services and products are in good health? When they detect a problem, how are they able to find the root cause in order to identify the causes of errors? When there is a fraud, how do they know what accounts were compromised? the answer is simple: *Log Analysis.*

For every event that occurs in the life of a WebApplication or Cloud Server, a line is registered under a *log file* describing every possible detail about it. After this, future analysis or reference is made to resolve any issue, predict another event or simply archived.

Log Analysis is the activity of analyzing log data to derive meaning from it. (Chuvakin, A., Schmidt, K., Phillips, C., & Moulder, P., 2013)

In the past this kind of analysis was, and still is, done using common unix tools, such as `cat(1)`, to print, `sed(1)`, stream editor, `grep(1)`, among others. (Basin, D., Schaller, P., & Schläpfer, M., 2011) These are used everyday by system administrators to get the information they are looking for inside the log files, stored in different locations depending on which kind of information they are dealing with.

Bernard J. Jansen in “Search log analysis: What it is, what’s been done, how to do it” (2006), defines that the process of analyzing logs has three major stages: Collection, Preparation and analysis.

- **Collection**: Defines what information one must collect in a transaction log.
- **Preparation**: the log is stored in a relational database, assigning the respective keys, cleaning the data to avoid errors, parsing it to create a standard.
- **Analysis**: SQL queries are created to generate the more useful information for the business.

According to Oliner, A., Ganapathi, A., & Xu, W. (2011) the common issues that come with the described log analysis stages are:

- Log Analysis is an ad-hoc process.
- Complexity increases with the different kinds of logs to parse.
- The logging process itself requires additional management.
- Only a few people know how to parse the logs.
- Logs are processed mostly once a day, which is pretty slow for certain decisions.
- Information to CEO’s is presented too late to take preventive actions.

In order to correct these problems a proposal and an implementation of a generic solution for the log analysis problem is presented in this research work.

For the different stages of the Log Analysis process the proposed solution looks as follows:

- **Collection**: LogStash capable of parsing the data in the tool itself.
- **Preparation**: ElasticSearch for storing and searching all the data.
- **Analysis**: Kibana is used to present the information to the decision makers in real time.

Through the following chapters we will analyze in depth the status of the practice in log analysis, present a detailed description of the proposed solution and show how it is being used in ABP and which have been the areas of impact so far. Finally we will comment on whether this strategy is suitable for other companies and in what context it could be used.

Chapter 2. Background

2.1 What is log analysis?

Operating Systems and applications come with mechanisms to report errors, warnings and security relevant events such as authentication. All these events are registered in *log files* to make events transparent and comprehensible. *The action of storing event messages is known as logging* and it can be used for behavioral research (Kennedy, J., 1983), leading to the optimization of services, improving security detecting and diagnosing security breaches and therefore explaining why things happened or predicting what is going to happen. (Basin, D., Schaller, P., & Schläpfer, M., 2011)

In practice, important messages often go undetected because of the large number of log entries triggered by irrelevant events all at the same time, and it is common that users or server administrators don't know where to search for specific log messages or how to configure them to obtain better results. (Basin, D., Schaller, P., & Schläpfer, M, 2011).

Log files often contain as many entries as events which, on their own, are meaningless. That's why it is important to correlate and filter these entries in order to summarize events and detect suspicious incidents. For instance, on a normal day, only one server in ABP generates around 7,000,000 lines [1]of different events. That's why different tools exist to work together with the system administrator to simplify the task of analyzing all the events.

2.1.1 Kinds of Log Messages

To differentiate between logs and simplify the search of events, there are different types of logs: (Chuvakin, A., Schmidt, K., Phillips, C., & Moulder, P., 2013)

- Informational
    - Designed to let users and administrators know that something benign has occurred.

[1] This number was calculated from the access of one of the servers in ABP.

- Debug
    - Generated from software systems in order to aid software developers troubleshoot and identify problems with code.
- Warning
    - Concerned with situations where things may be missing or needed for a system but will not impact system operation.
- Error
    - To relay errors that occur at various levels in a computer system.

Vaarandi, R. (2005) mentions how UNIX tools like `*grep*`, `*sed*` or `*find*` have been used for years to filter and process information that matters for companies. Tools relying on regular expressions and used only by the system administrator or the data analyzer, familiarized with the syntax. Of course, results are only reachable by them and it is far from real time monitoring.

Solano, J., & Leiva, E. (2014) suggest that the information generated by organizations characterizes itself by the speed to which it is produced, bringing with it a development of new solutions to process the information as quickly as possible. That's why, some companies need to extract information in real time. One of the highly recommended tools for real time processing[2] is Apache Spark.

Log files increase their size rapidly, but that's not the main problem. Another problem with log analysis is that normally people in the company have to wait for the data analyst to run the queries and deliver the results. About a year ago, (Harris, D., 2015), AirBnb, a community marketplace where guests can book spaces from hosts, open sourced *Airpal*, a tool based on Presto[3], to allow the employees across divisions and roles to get fast access to their data rather than having to wait for a data analyst or data scientist to run a query for them. Now they have access to:

- A visual interface
- Previews of the data they're accessing
- Ability to share and reuse queries.

[2] Cloudera, 2015 http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/spark.html
[3] https://prestodb.io/

Nowadays there are several ways to understand better the logs that each company is generating.

2.2 Requirements of a Log Analysis Tool

As Cid, D. (2006) described, Log files easily reach high amounts of volume, registering every event that happens on every server and since each type of log file has a different format, the complexity increments and makes the analysis of these files more complicated as they grow. That's why automatic log analysis is essential.

Any log analysis tool should do at least the following:

- Understand your logs.
- Look for patterns:
    - Correlate bad events to indicate attack or intrusion
    - Correlate good events with bad events
    - Correlate good events (eg. too many successful logins for the same user across multiple hosts in a small period of time)
    - Look for unusual patterns. (eg. Too many requests in a normal day)

2.3 Tools for Log and Data Analysis

2.3.1 Apache Spark

(http://spark.apache.org)

Apache spark is a fast and general engine for large-scale data processing. It's used by companies like Amazon, eBay (for log transaction aggregation and analytics), Ooyala (fast queries), Shazam, Yahoo and others. It performs batch processing (similar to MapReduce) and new workloads like streaming, interactive queries and machine learning. (Spark.apache.org,. 2015)

Spark has been used to sort 100 TB of data 3X faster than Hadoop MapReduce on 1/10th of the machines and the largest cluster known has over 8000 nodes (it is high scalable). It doesn't need Hadoop to run, but some form of shared file

system is required. It uses fast Remote Procedure Calls for task dispatching and scheduling and threadpool for the execution of tasks, as well as checkpoint-based and lineage-based recovery to make it fault tolerant and increase speed. (Spark.apache.org, 2015)

2.3.2 Apache Hive

(http://hive.apache.org)

The Apache Hive data warehouse software facilitates querying and managing large datasets residing in distributed storage. Hive provides a mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL. At the same time this language also allows traditional map/reduce programmers to plug in their custom mappers and reducers when it is inconvenient or inefficient to express this logic in HiveQL. (Hive, 2015)

2.3.3 Dremel

(http://research.google.com/pubs/pub36632.html)

To improve processing of data, Google created Dremel, interactive ad-hoc query system for analysis of read-only nested data (Web-Scale Data Sets). It uses the MapReduce paradigm to process queries faster. (Melnik, S., Gubarev, A., Long, J., Romer, G., Shivakumar, S., Tolton, M., & Vassilakis, T., 2011)

2.3.4 MapReduce

(http://goo.gl/3WV7jM)

*MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. (*Dean, J., & Ghemawat, S., 2008*)*

A typical MapReduce computation processes many terabytes of data on thousands of machines. Programmers find the system easy to use: hundreds of

MapReduce programs have been implemented and upwards of one thousand MapReduce jobs are executed on Google's clusters every day.

2.3.5 Presto

(http://prestodb.io)

Prestodb.io,. (2015) and Traverso (2013), define this tool as a distributed SQL query engine for Big Data optimized for ad-hoc analysis at interactive speed:

- Run interactive analytic queries wherever the data is stored (currently supports Hive, Cassandra, relational databases or even proprietary data stores)
- Includes complex queries, aggregations, joins, and window functions.

2.3.6 ACW

(http://aws.amazon.com/cloudwatch/)

Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. You can use Amazon CloudWatch to:

- Collect and track metrics
- Collect and monitor log files
- Set alarms.
- View graphs and statistics

2.3.7 Suro

(https://github.com/Netflix/suro)

Used by Netflix, based in Apache Chukwa, and it provided benefits in integration such as:

- Supports arbitrary data formats. Users can plug in their own serialization and deserialization code
- Supports dispatching events to multiple destinations with dynamic configuration

- Supports configurable store-and-forward on both client and collector
- Batch Processing generated by Hadoop Jobs
- Real Time Computation
- Collector of events in detail.

2.3.8 Sumo Logic

(https://www.sumologic.com/)

The Sumo Logic service is powered by patent-pending Elastic Log Processing™ and LogReduce™ (to filter out the noise in the data) technologies, and it provides the following capabilities:

- Collect and centralize data
- Search and analyze
- Detect and predict anomalies.
- Monitor and Visualize the data.
- Alert and Notify

Sumo Logic enables Netflix to realize instant cost savings, gain unprecedented application insight, and monitor and troubleshoot their cloud and on-premise IT infrastructure and applications in real-time.

2.4 Companies doing Log Analysis and their Approaches

2.4.1 Google

(http://www.google.com/about/)

Google is one of the first companies that developed a new way to deal with high amounts of data. They were processing around 24 petabytes of data per day (+ 24 000 000 000 megabytes or + 1,000,000,000,000,000 bytes). They had the data, the processing power and statistical know-how to come with a new solution for better results. They named it MapReduce. (Mayer-Schönberger, V., & Cukier, K. 2013.)

*Google's method [...] was built on "big data", the ability of society to harness information in novel ways to produce useful insights or goods and services of significant value. (*Mayer-Schönberger, V., & Cukier, K., 2013*)*

But processing the data is not enough. It is important to understand what is behind it and to visualize the data. That's why companies like Facebook and Netflix develop their own solutions according to their business needs..

2.4.2 Facebook

(https://www.facebook.com/facebook/info?tab=page_info)

Traverso (2013) describes how *Facebook's warehouse data is stored in a few large Hadoop/HDFS-based clusters. Hadoop MapReduce and Hive are designed for large-scale, reliable computation, and are optimized for overall system throughput.* The data is used for a wide range of applications, from traditional batch processing to graph analytics, machine learning, and real-time interactive analytics.

Facebook is a Data Driven company, and with Presto, they are capable of handling over 300 Petabytes of daily data, designed to process queries for data with lower latency, or as they say, quicker from an end-user standpoint. Presto, developed in Java, dynamically compiles certain portions of the query plan down to byte code, which lets the JVM optimize and generate native machine code. (Adweek, 2013)

They extend Presto's functionality constantly and improve performance using Apache Hive as well. Netflix got a lot of help when Facebook open sourced this tool and they use it since making contributions too as Cohen, D. (2013) said, making a better tool each day.

2.4.3 Netflix

(https://pr.netflix.com/WebClient
/loginPageSalesNetWorksAction.do?contentGroupId=10476)

Orzell, G. (2012) comments how Netflix uses Servo, a library designed to make it easy for developers to export metrics from their application code, register them with JMX, and publish them to external monitoring systems such as Amazon's CloudWatch. With this they keep track of the metrics they need to make their predictions and suggestions. To make the analysis of the data they collect, Sumo was the choice as Sumo Logic,. (2015) described.

Netflix's developers use a variety of tools in the Hadoop ecosystem. In particular:

- Hive for ad hoc queries and analytics
- Pig for ETL(Extract, Transform, Load) and algorithms.
- Vanilla java-based MapReduce is also occasionally used for some complex algorithms.
- Python is the common language of choice for scripting various ETL processes and Pig User Defined Functions (UDF).

Hadoop clusters are accessible via a number of "gateways", which are just cloud instances that running jobs of Hadoop, Hive and Pig.

Table 1: Log Analysis Tools and Companies that use them

	Google	Facebook	Netflix
MapReduce	X	X	X
Hive		X	X
Spark			
Hadoop		X	X
Python		X	X
Presto		X	X
ACW			X
Dremel	X		

2.5 AdBlock Plus Log Analysis

ABP's main goal is to show how an acceptable advertisement should be on the internet, filtering the rest of the annoying

ads. It also blocks malware, tracking cookies or other things people don't want in their browsers.

AdBlock Plus is the most downloaded extension in the world with over 300 million downloads. So ABP's log analysis needs are in line with those of the previously mentioned companies.

ABP promises it will never collect any of your personal data. And they keep their promise. Every data in the servers is erased after 30 days, top. The first thing that is deleted are Log Files, which makes prediction and calculations about users and impact quite difficult.

The data that comes to the servers is anonymized[4] when possible, used to generate usage statistics as well as to investigate potential security issues and forum/blog spam. The detailed logs are retained for a period of 30 days after which only the aggregated usage statistics remain.

In average, 75 GB of Log Files are stored in ABP servers every day. Also, daily, the Data Analyst (an actual job title within ABP) run his R scripts to aggregate the important information. Before doing this, the data is collected and cleaned of course with his personal tools such as `grep`, makefiles, and programming. The tests to process the information must be done with a small size of the total amount of the logs. The complete analysis takes around 20 minutes processing. Running everything in his RAM memory.

The Data Analyst, presents the valuable information every monday or upon request, scheduling a meeting with the people in the company that need any information about ABP.

The DevOps team and the Sysadmin in the company have the responsibility of monitoring the servers and currently UNIX is saving the day once more. Tools such as `htop`, `sed`, `tail` and more are used to detect anomalies that need to be detected by humans, nevertheless this can be improved into a more automated way.

[4] To remove any information that shows which particular person something relates to. (Cambridge Advanced Learner's Dictionary & Thesaurus, 2015)

2.5.1 Disadvantages

The current way of doing Log Analysis in ABP has the following disadvantages:

- Data is processed only once a day. Which generates delays in taking decisions.
- Data is presented to decision makers only once a week.
- Creating new analysis is an operation that only one person knows how to perform.
- Data and access manipulation only available for one person

Chapter 3. Proposed Solution

Based on the disadvantages mentioned in the previous section, ABP needed a better way to analyze their logs and monitor their systems. The new solution should fulfill the following requirements.

3.1 Requirements

In order to make sure that this system is going to help users achieve the business objectives, the requirements have been splitted into two categories with the following characteristics:

Table 2. Characteristics of the requirements

ID	● FR -> Functional Requirement ● NFR -> Non Functional Requirement The ID is followed by a number to identify every requirement.
Name	A word to identify the requirement
Description	Explanation of the requirement
Priority	Importance under development. (High, Medium, Low)
Flexible	Possibility that the requirement changes.
Verifiable	Easiness to corroborate the requirement exist.
Necessary	● Yes -> It is a must. ● Optional -> It would be nice if the requirement is there.

3.1.1 Functional requirements. What should the system do?

Table 3. FR-1

ID	FR-1

Name	Authentication
Description	Permit login to authorized users.
Priority	High
Flexible	No
Verifiable	Easy
Necessary	Yes

Table 4. FR-2

ID	FR-2
Name	Visualization
Description	Display general information about the company (from the system logs).
Priority	High
Flexible	No
Verifiable	Easy
Necessary	Yes

Table 5. FR-3

ID	FR-3
Name	Dashboarding
Description	Show personalized dashboards for each area in the company.
Priority	High
Flexible	No
Verifiable	Easy
Necessary	Yes

Table 6. FR-4

ID	FR-4
Name	Searching
Description	Search for specific information according to the user needs.
Priority	High
Flexible	No
Verifiable	Easy
Necessary	Yes

Table 7. FR-5

ID	FR-5
Name	Aggregation
Description	Let the user aggregate information.
Priority	High
Flexible	No
Verifiable	Easy
Necessary	Yes

Table 8. FR-6

ID	FR-6
Name	Medic
Description	Display health & status of the servers.
Priority	High
Flexible	No
Verifiable	Easy

Necessary	Yes

3.1.2 Non-Functional requirements.

Table 9. NFR-1

ID	NFR-1
Name	Accesibility
Description	The system should be accessible for all the members in the headquarters.
Priority	Medium
Flexible	Yes
Verifiable	Easy
Necessary	Yes

Table 10. NFR-2

ID	NFR-2
Name	Reliability
Description	The system will be available 99% of the time unless failures occur.
Priority	Medium
Flexible	Yes
Verifiable	Medium
Necessary	Yes

Table 11. NFR-3

ID	NFR-3
Name	Tolerant
Description	The system will be distributed and fault-tolerant.
Priority	Medium
Flexible	No
Verifiable	Easy
Necessary	Yes

Table 12. NFR-4

ID	NFR-4
Name	Scalable
Description	To achieve scalability admins will be able to add servers at will.
Priority	Medium
Flexible	No
Verifiable	Easy
Necessary	Yes

Table 13. NFR-5

ID	NFR-5
Name	Open
Description	All the architecture, systems and functionality will be open source.
Priority	High
Flexible	No
Verifiable	Easy

Necessary	Yes

Table 14. NFR-6

ID	NFR-6
Name	Portable
Description	It will be reusable and portable enough (in a puppet manifest) to be implemented in any circumstance.
Priority	High
Flexible	No
Verifiable	Easy
Necessary	Yes

Table 15. NFR-7

ID	NFR-7
Name	Trustworthy
Description	High reliability due to the previously cleaned and parsed logs.
Priority	High
Flexible	No
Verifiable	Medium
Necessary	Yes

Table 16. NFR-8

ID	NFR-8
Name	Secure

Description	Secure enough to permit access only to the people in the company.
Priority	High
Flexible	No
Verifiable	Easy
Necessary	Yes

Table 17. NFR-9

ID	NFR-9
Name	Friendly
Description	Very user friendly thanks to the last part of the architecture, Kibana. Explained in detail below.
Priority	High
Flexible	No
Verifiable	Easy
Necessary	Yes

3.2 Benefits

With the system, the following benefits will be achieved:

- Automate the way data is interpreted as much as possible.
- Make aggregations even easier for the people inside the company (In 2009, Facebook counted 29% of its employees (and growing!) as Hive users. More than half (51%) of those users are outside of Engineering according to Facebook's data analysis center).
- The monitoring of the servers would be simplified to a more visual and interactive way, providing:
    - The ability of finding patterns with usage behaviour.
    - Alerts and more advantages are pursued with the ELK solution.

- Faster reaction to anomalies.
- Better insight.
- More specific interpretation to the behaviour of the users across the world.

3.3 The solution: 3 steps

The core of the solution resides on being capable of understanding the logs and obtaining the information we need from them, transforming a lot of senseless information, into defined information with enough context to focus in the real needs of the company.

This solution is divided into three steps:

3.3.1 Centralize and parse the data.

A lot of variables can be taken into account, but the main issue remains in reducing the gap that the decentralized logs create, each one of these contain important information about the company. This Information should be known by the heads of the company, improving the decision capacity and creating the opportunity for future predictions.

Centralized logging can be very useful when attempting to identify problems in servers or applications, giving the possibility to search through all of the logs in a single place.

A lot of times corrupt data is received and recorded into log files, producing inconsistent or not reliable information or even breaking the analysis scripts. That’s why parsing and cleaning data is important.

Parsing can be done before or after centralizing the data, it depends on the business needs.

3.3.2 Store important information and make it traceable.

When all the data is cleaned and parsed, it is ready to be indexed and stored. With this, the possibility for searching through all our data is a reality for every

person in the company, providing the answers they need or even using that information in a different fashion.

3.3.3 Aggregate the data and present it.

With the data indexed and fully searchable, the next step is to interpret it and transform it into valuable information. This will bring better insight into the company and now predictions can be made bringing statistics into the game. Also issues can be found analyzing data from the past and bottlenecks could be found as well. When this information is revealed it is easier to come with a solution and put it to the test.

Aggregating data to transform it into information is not enough. Normally this is responsibility of the person who analyze the data. This person should come up with ideas of mixing and filtering data to present valuable information hidden under more information (data mining[5]). Once this task is done, sharing is important because people without IT background is just asking questions and hoping that the data analyst will bring an answer.

Business people do not mess with raw data, that's why it is better to present the information to them in the most straightforward way as possible. If they can access to the results of the questions they always ask is much better for them, this way they can keep track of useful information depending on their department (PR, Client communication, directive decisions and so on) improving their performance.

The proposition is that the combination of the three previous steps will create a killer combo to keep better track of the data inside the company.

[5] Data mining is the practice of automatically searching large stores of data to discover patterns and trends that go beyond simple analysis. (Docs.oracle.com, 2015)

Chapter 4: Implementation of the Solution

For a detailed description of the solution, please refer to the Appendix A: Implementation Log.

In this chapter we provide a 10,000 ft overview of the implemented architecture.

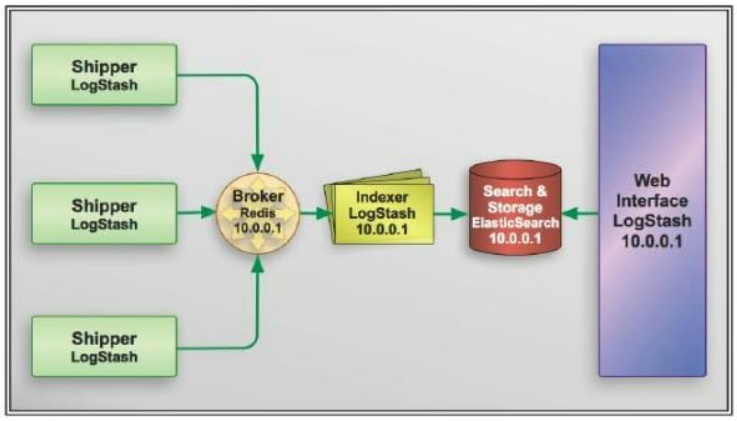
4.1 Tools Used

4.1.1 Logstash - Collect, Parse, & Enrich Data

Logstash is a data pipeline that helps to process logs and other event data from a variety of systems. With it we can Collect, Parse, & Enrich the Data.

Is an Open Source tool for managing events and logs under the licence Apache 2.0. You can use it to collect logs, parse them, and store them for later use, it is written in JRuby and runs in a Java Virtual Machine (JVM).

Its architecture, shown below, is message-based and, *rather than separate agents or servers, Logstash has a single agent that is configured to perform different functions in combination with other open source components.* (Turnbull, 2013)



[Figure 1. The logstash architecture]

- Shipper sends events to logstash.
- Broker and indexer receives and index events.
- Search and storage: Allows to search and store events.
- Web Interface: Web based interface.

The logstash servers run independently which allow to separate components and scale Logstash.

4.1.2 Elasticsearch - Search & Analyze Data in Real Time

Elasticsearch is a search and analytics engine. Thanks to it almost any action can be performed using a simple RESTful API using JSON over HTTP. It is easy to scale it supports advanced search features and indexing.

Elasticsearch is an open source search engine with a RESTful web interface and schema-free JSON documents written in Java. It is a tool that facilitates storing, searching and querying written words. ES is a standalone database server that takes data in and stores it in a format optimized for language based searches supporting Multitenancy. Due to the number of shards where the data is stored, the replication of them and the resilient clusters, it makes of ES a reliable, asynchronous, distributed and highly available tool, providing as well real time search.

It is able to achieve fast search responses because, instead of searching the text directly, it searches an index instead in a distributed way. This type of index is called an inverted index, because it inverts a page-centric data structure (page->words) to a keyword-centric data structure (word->pages).

To represent the data, Elasticsearch organizes the data in documents. Each one of them defined by an index and contains different fields.

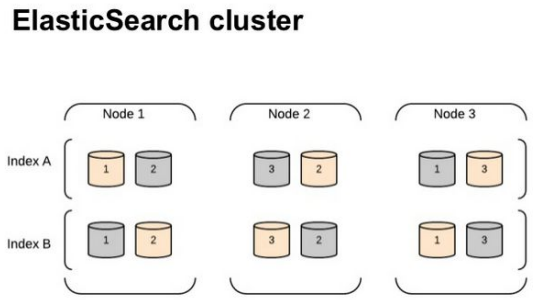
```
{
  "ok": true,
  "status": 200,
  "name": "Psyche",
  "version": {
    "number": "0.90.0",
    "snapshot_build": false
  },
  "tagline": "You Know, for Search"
}
```

[Figure 2. ES document]

It is built in top of Apache Lucene, a Java library search engine that implements the inverted indexing, making queries faster. Nevertheless, ES provides to it more benefits that makes this technology fully scalable:

- A simpler API.
- Interoperation with non-Java/JVM languages.
- Operational ease of use.
- Clustering and replication.
- Good defaults for complex Lucene classes.

[Figure 3. Lucene Inverted Index structure]



[Figure 4. Elasticsearch architecture]

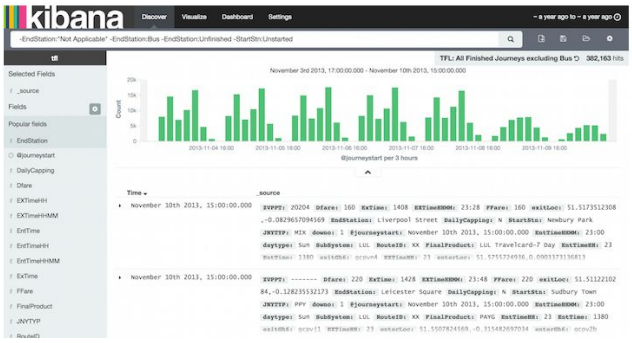
4.1.3 Kibana - Explore & Visualize Data

is the elasticsearch visualization engine. It allows dynamic interaction with the data.

Kibana is the open source analytics and visualization platform to search, view, and aggregate data stored in Elasticsearch. Thanks to it we can give shape to the data the logs provide, which brings a better understanding to big amounts of data. The code can be found in:

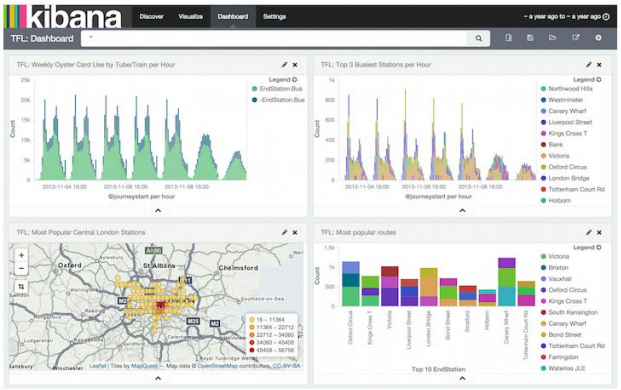
https://github.com/elastic/kibana

It is made of pure HTML and Javascript, this means that there is no server side, facilitating the implementation of it.



[Figure 5. Data discovery]

The data can be visualized and analyzed using a variety of charts, tables and maps, giving the possibility to create different dashboards to align with the company needs.



[Figure 6. Kibana Dashboards]

All of the tree are open source projects and the code can be found on GitHub.

4.1.4 Puppet - Automation Makes IT Better

Puppet is a widely adopted configuration management system across every major market with over 10 million nodes managed (Puppet Labs, 2015) and it is used mainly for discovering, configuring, and managing IT infrastructure. It is an automated administrative engine that can be used in different environments such as Linux, Unix, and Windows systems. It performs administrative tasks (such as adding users, installing packages, and updating server configurations) based on a centralized specification.

Puppet Labs,. (2015) explains us that *Puppet is a configuration management system that allows you to define the state of your IT infrastructure, then automatically enforces the correct state.* Puppet automates tasks that sysadmins often do manually, freeing up time and mental space so sysadmins can work on the projects that deliver greater business value. Puppet ensures consistency, reliability and stability.

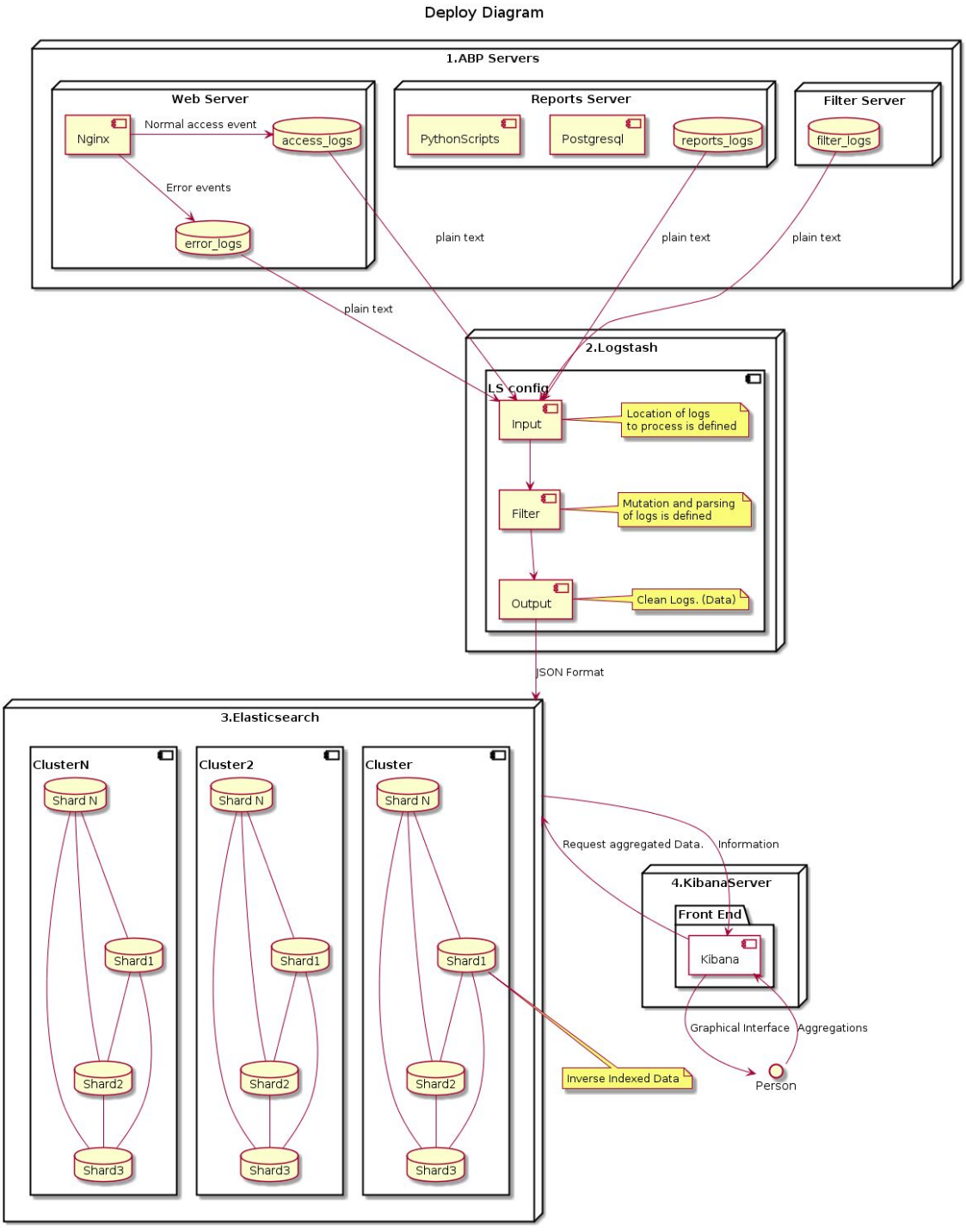
Puppet was first created in 2005 as an open source configuration management tool, and is available for free download under the Apache 2.0 license. As a declarative, model-based approach to configuration management, open source Puppet lets you define the desired state of your infrastructure — rather than how to get there — using the Puppet language. Once you've defined your desired state, Puppet continually ensures it stays that way, enforcing the correct configurations and making sure the right services are up and running.

4.1.4.1 Why use Puppet?

Olindata.com,. (2015) highlights four main benefits:

- **Productivity / Efficiency** - Most IT management solutions deliver efficiency of 20-30 nodes per sysadmin. Puppet enables 100s and even 1000s of nodes per sysadmin.
- **Responsiveness To Business Needs** - Using Puppet, customers have dramatically reduced the time it takes them to deliver applications into production, from weeks to days and even hours.
- **Eliminate Configuration Drift** - With Puppet, your nodes (servers, desktops, etc.) remain in the state you set for them, dramatically improving service availability, reliability, scalability, and performance.
- **Visibility** - Puppet provides rich data sets not only of infrastructure configuration but also of any changes to that infrastructure, whether under direct control of Puppet or not. Much more visibility is acquired into the occurring changes in your infrastructure over time and their impact to service levels.

4.2 Deploy Diagram



[Diagram 1: ELK stack deployment diagram]

Above, in Diagram 1, the involved components are described and numbered.

The system behaves as follows:

1. The **ABP Servers** work normally. Doing the functions expected from them: serving Web Services, files, filtering information, etc. These servers send their log data through a TCP Port.

2. **Logstash** server reads the log data and executes 3 main tasks:

    a. **Input**. Describes where is the log information coming from, or where can it be accessed.

    b. **Filter**. The Parsing logic must be defined using Regular Expressions and tools provided by logstash to gather only useful data, transforming it into JSON format to provide better organization of the data, as well as variable types and so on, facilitating the latter process.

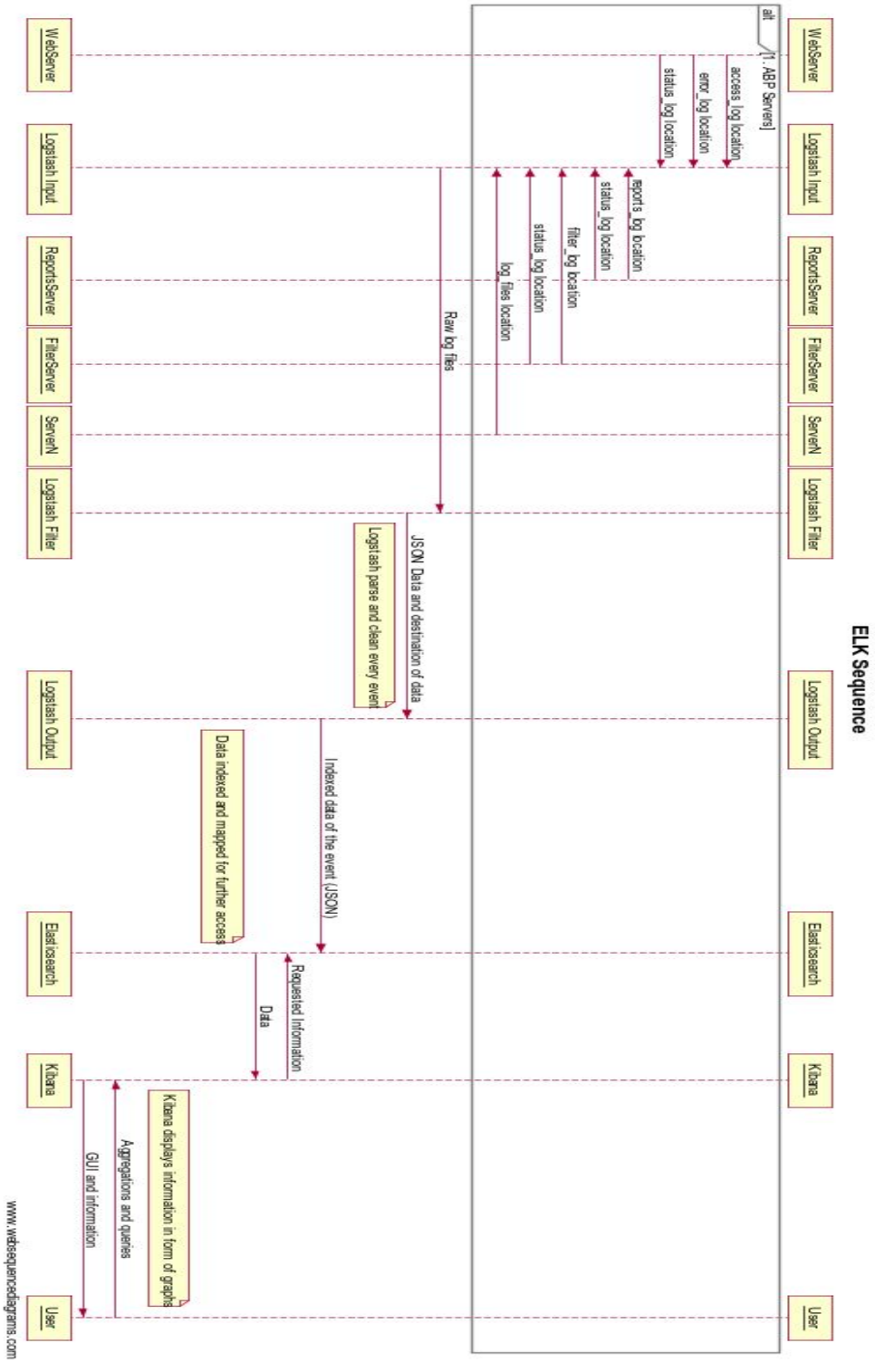
    c. **Output**. Send the parsed and cleaned data to Elasticsearch to be inverse indexed and stored for further access.

3. **Elasticsearch** receives the data and inverse indexes it in a previously configured cluster providing redundancy and failure tolerability, writing the information across multiple shards, each one of them containing different mappings[6] depending on the gathered information.

4. The **Kibana server** is the GUI of the system, providing a friendly environment for the user to create aggregations to create information of the queried data.

[6] Mapping is the process of defining how a document should be mapped to the Search Engine, including its searchable characteristics such as which fields are searchable and if/how they are tokenized. A way to divide the documents in an index into logical groups. Think of it as tables in a database. (Elasticsearch, 2015)

4.3 Sequence Diagram



[Diagram 2: Sequence diagram of the solution]

Above, in Diagram 2 sequence of the system is represented.

The Logstash server should be configured to recognize all the new incoming logs, parse them and store the valuable information into Elasticsearch.

Thanks to the different plugins that come with Logstash is easy to keep track of the logs from different inputs such as TCP/UDP, files, syslog, STDIN and other sources. It is not likely to find an environment where the possibility to extract logs doesn't exist.

When the logs hit the Logstash server, the configured filters allow to modify, manipulate and transform the events, extracting the information that is needed and give context to it.

After this happens, Logstash supports a variety of ways of outputting the data, including TCP/UDP, email, files, HTTP, Nagios[7] and more online services.

Alerts, graphs, storage can be also integrated.

The information is stored under Elasticsearch. How? It is indexed word per word using Inverted Indexing, obtaining with this better results when a search is made. It increases speed of the queries.

This technique is used among multiple search engines.

Once the information is stored in Elasticsearch, it can be accessed from Kibana, the GUI of the system providing a more friendly environment for the user, since aggregations and different kind of graphics can be made using this interface (or with code for more advanced users) presenting useful information which can be organized in different dashboards and can be monitorized live.

The automation of this combination of tools is very important, saving a lot of time, energy and configurations, as well as providing code that acts as documentation for the system, adding to it scalability, redundancy and ease of maintenance. A puppet manifest was created and open sourced. It can be accessed in:

https://github.com/AAlvz/infrastructure/tree/elasticsearch

[7] Nagios monitors your entire IT infrastructure to ensure systems, applications, services, and business processes are functioning properly. (Nagios, 2015)

https://github.com/AAlvz/logstash

This architecture allowed us to obtain the benefits expected in chapter 2. The actual installation and configuration is a lot harder. In the following chapter we explore the results of this implementation.

Chapter 5. Results

In this chapter are described the results of the experiment, emphasizing the methods used before, their difficulty and their results compared with the suggested solution, how complicated it is and the results acquired divided in the main phases of the Analysis Process.

Table 18: Comparison of the current state vs prototype implementation.

Analysis Process	How it was done before?	With the proposed solution	Main Benefits
Data Parsing	● R scripting ● Once a day ● Takes around 20 minutes to finish (using RAM)	● Logstash ● Always active	● Fast reaction ● Information always updated
Data Storing	● PostgreSQL	● Elasticsearch	● No SQL syntax ● Live storing ● Facilitate searching
Visualization	● Excel Tables ● R graphics	● Kibana	● Standardized ● Self updating ● Friendly ● Allows aggregations

5.1 Data Parsing

5.1.1 Current Solution

The first thing that should be done is execute a Makefile done by the Data Analyst to:

- Gather the log files from the different servers
- Remove unnecessary characters from the log lines such as quotes or symbols

- Separates the cleaned logs and the original ones into different files
- Runs R scripts to:
    - Process data and update values
    - Aggregate processed files
    - Make additional output
- Diverse minor tasks

Then, to parse requests made to ABP servers, the main script is made on python. Its main tasks are:

- Split the log line into fields.
- Remove corrupt data
- Split some fields to get more fields
- Get values from fields
- Standardize values if necessary.
- Output the same log in a clean fashion.

Mainly, the objective of the last executable files is to clean the logs and get them ready to be processed and get information out of them.

The Makefile by itself has around 230 lines of code.
One of the main log processors files (in R) has around 550 lines of code, added to another one containing 150 lines of code. More than 20 R files (around 1900 lines of code) are used to parse the data and get information out of it.

5.1.2 Proposed Solution

With the prototyped solution all those files got replaced with only a LogStash of with 20 lines of code (See Appendix B).

Jordan Sissel (2012), the Logstash creator claims that having troubles writing Logstash code is a Logstash bug. According to him, Logstash is supposed to be easy to use, understandable, and self documented.

5.2 Storing data

5.2.1 Current Solution

The connections to the database are handled by a python script and invoked every time data is added, updated or even erased, therefore several code snippets to communicate with the database are all around different scripts. If this continues growing and is not standardized, it could get more messy.

5.2.2 Proposed Solution

With the **output** implementation of Logstash, the connections can be handled automatically by this tool even giving the possibility of having different outputs (even one for the corrupted data which can be useful as well) handled in a central solution adding this small part to the config file:

```
output {
  if "_grokparsefailure" in [tags] {
    file {
      path  => "<%= @log_dir %>/logstash-grokparsefailure.log"
      codec => line
    }
  } else {
    elasticsearch {
      host     => localhost
      protocol => "http"
    }
  }
}
```

[Code 1. Logstash output]

What happens with the filter and output section of Logstash can be better seen here:

Here are 2 lines of access_logs of nginx for the filter1 server.

```
xxx.xxx.xxx.xxx - - [17/Nov/2014:23:59:23 +0000] "GET /easylist.txt?_=1415911744040 HTTP/1.1" 200 398623 "-" "Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.111 Safari/537.36" "-" https "en-US,en;q=0.8" "easylist-downloads.adblockplus.org" "AdBlock/2.13.2"
```

```
xx.xxx.xx.xx - - [17/Nov/2014:23:59:23 +0000] "GET /antiadblockfilters.txt?addonName=adblockplus&addonVersion=2.6.6&application=firefox&applicationVersion=33.1&platform=gecko&platformVersion=33.1&lastVersion=201411161830 HTTP/1.1" 200 1839 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0" "-" https "en-US,en;q=0.5" "easylist-downloads.adblockplus.org" "-"
```

[Code 2. Log lines standard]

This is indexed into this JSON:

```
{
        "message" => "128.195.202.154 - - [17/Nov/2014:23:59:23 +0000] \"GET /easylist.txt?_=1415911744040 HTTP/1.1\" 200 398623 \"-\" \"Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.111 Safari/537.36\" \"-\" https \"en-US,en;q=0.8\" \"easylist-downloads.adblockplus.org\" \"AdBlock/2.13.2\"",
       "@version" => "1",
     "@timestamp" => "2014-11-17T23:59:23.000Z",
           "host" => "debian",
           "path" => "/home/aalvz/vagrant/logstash/logs/test_logs",
           "type" => "test",
       "clientip" => "dc45a135a41c09135eda88ad6e0214b7698b7501",
          "ident" => "-",
           "auth" => "-",
      "timestamp" => "17/Nov/2014:23:59:23 +0000",
           "verb" => "GET",
        "request" => "/easylist.txt?_=1415911744040",
    "httpversion" => "1.1",
       "response" => "200",
          "bytes" => "398623",
          "agent" => "\"Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.111 Safari/537.36\"",
  "xforwardedfor" => "\"-\"",
           "lang" => [
      [0] "\"en-US",
      [1] "en;q=0.8\""
  ],
        "hostabp" => "easylist-downloads.adblockplus.org",
       "adblock" => "\"AdBlock/2.13.2\"",
         "geoip" => {
                  "ip" => "128.195.202.154",
       "country_code2" => "US",
       "country_code3" => "USA",
        "country_name" => "United States",
      "continent_code" => "NA",
         "region_name" => "CA",
           "city_name" => "Irvine",
         "postal_code" => "92697",
            "latitude" => 33.640299999999996,
```

```
            "longitude" => -117.76939999999999,
             "dma_code" => 803,
            "area_code" => 949,
             "timezone" => "America/Los_Angeles",
       "real_region_name" => "California",
              "location" => [
          [0] -117.76939999999999,
          [1] 33.640299999999996
       ]
   },
        "resource" => "/easylist.txt",
       "lang_real" => "\"en-US",
        "ip_agent" => "dc45a135a41c09135eda88ad6e0214b7698b7501\"Mozilla/5.0 (Windows NT
5.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.111 Safari/537.36\"",
            "name" => "Chrome",
              "os" => "Windows XP",
         "os_name" => "Windows XP",
          "device" => "Other",
           "major" => "38",
           "minor" => "0",
           "patch" => "2125",
               "_" => "1415911744040"
}
{
            "message" => "64.203.40.36 - - [17/Nov/2014:23:59:23 +0000] \"GET
/antiadblockfilters.txt?addonName=adblockplus&addonVersion=2.6.6&application=firefox&applic
ationVersion=33.1&platform=gecko&platformVersion=33.1&lastVersion=201411161830
HTTP/1.1\" 200 1839 \"-\" \"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101
Firefox/33.0\" \"-\" https \"en-US,en;q=0.5\" \"easylist-downloads.adblockplus.org\" \"-\"",
            "@version" => "1",
          "@timestamp" => "2014-11-17T23:59:23.000Z",
               "host" => "debian",
               "path" => "/home/aalvz/vagrant/logstash/logs/test_logs",
               "type" => "test",
           "clientip" => "c1a6065ec204155b53b1c7b28616c1342e7c21aa",
              "ident" => "-",
               "auth" => "-",
          "timestamp" => "17/Nov/2014:23:59:23 +0000",
               "verb" => "GET",
            "request" =>
"/antiadblockfilters.txt?addonName=adblockplus&addonVersion=2.6.6&application=firefox&applic
ationVersion=33.1&platform=gecko&platformVersion=33.1&lastVersion=201411161830",
        "httpversion" => "1.1",
           "response" => "200",
              "bytes" => "1839",
              "agent" => "\"Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101
Firefox/33.0\"",
       "xforwardedfor" => "\"-\"",
               "lang" => [
       [0] "\"en-US",
       [1] "en;q=0.5\""
   ],
            "hostabp" => "easylist-downloads.adblockplus.org",
```

```
            "adblock" => "\"-\"",
              "geoip" => {
                  "ip" => "64.203.40.36",
        "country_code2" => "US",
        "country_code3" => "USA",
         "country_name" => "United States",
       "continent_code" => "NA",
         "region_name" => "CA",
           "city_name" => "Fountain Valley",
         "postal_code" => "92708",
           "latitude" => 33.71000000000001,
          "longitude" => -117.9478,
           "dma_code" => 803,
          "area_code" => 714,
           "timezone" => "America/Los_Angeles",
     "real_region_name" => "California",
           "location" => [
        [0] -117.9478,
        [1] 33.71000000000001
      ]
  },
          "resource" => "/antiadblockfilters.txt",
         "lang_real" => "\"en-US",
          "ip_agent" => "c1a6065ec204155b53b1c7b28616c1342e7c21aa\"Mozilla/5.0 (Windows
NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0\"",
               "name" => "Firefox",
                 "os" => "Windows 7",
            "os_name" => "Windows 7",
             "device" => "Other",
              "major" => "33",
              "minor" => "0",
          "addonName" => "adblockplus",
       "addonVersion" => "2.6.6",
        "application" => "firefox",
  "applicationVersion" => "33.1",
          "platform" => "gecko",
     "platformVersion" => "33.1",
        "lastVersion" => "201411161830"
}
```


[Code 3. JSON Parsing]

The output section of Logstash takes care of storing the information in different mechanisms such as csv, a file, stdout, elasticsearch, rabbitmq, rackspace, redis, a websocket and more, the point is that when the data is indexed out of the logs, it's easier to handle it.

5.3 Data Queries and access

5.3.1 Current Solution

Talk to the data analyst to get the information you need. He will then, create an aggregation using R and Python scripts and then process the logs he has so far to get an accurate result.

5.3.2 Proposed Solution

Now that all the data is stored in Elasticsearch, it is easy accessible (for authorized people) thanks to the elasticsearch RESTful API and it will be a real time query.

The same storage method that Elasticsearch uses gives the possibility to organize better the information in a understandable way using the index that it's provided:

Use the /{index}/{type}/{id}/_source endpoint to get just the _source field of the document.

Here some examples:

- The search API allows to execute a search query and get back search hits that match the query.
    - curl -XGET localhost:9200/_search?q=user:kibana
    - Returns every match for the ‘kibana’ user in all indexes

- Get a list of the indexes stored at the moment.
    - curl localhost:9200/_cat/indices?v

- Get all the visualizations stored in the ‘kibana’ index
    - curl -XGET localhost:9200/.kibana/visualization/_search?

- Delete everything
    - curl -XDELETE 'http://localhost:9200/*/'

Before they needed to talk to the data analyst to get any kind of information, now they can use the API.

5.4 Presenting data

5.4.1 Current Solution

Every monday there is a meeting to present all the relevant advances in the different fields in the company. Until now, the presented data always changes, looking for the best way to present it to the CEO, CTO and Stakeholders, always trying to answer their questions with data analyzed in the weekend from the previous week. Different graphs are presented every week and new questions (and not so new) always rise.

5.4.2 Proposed Solution

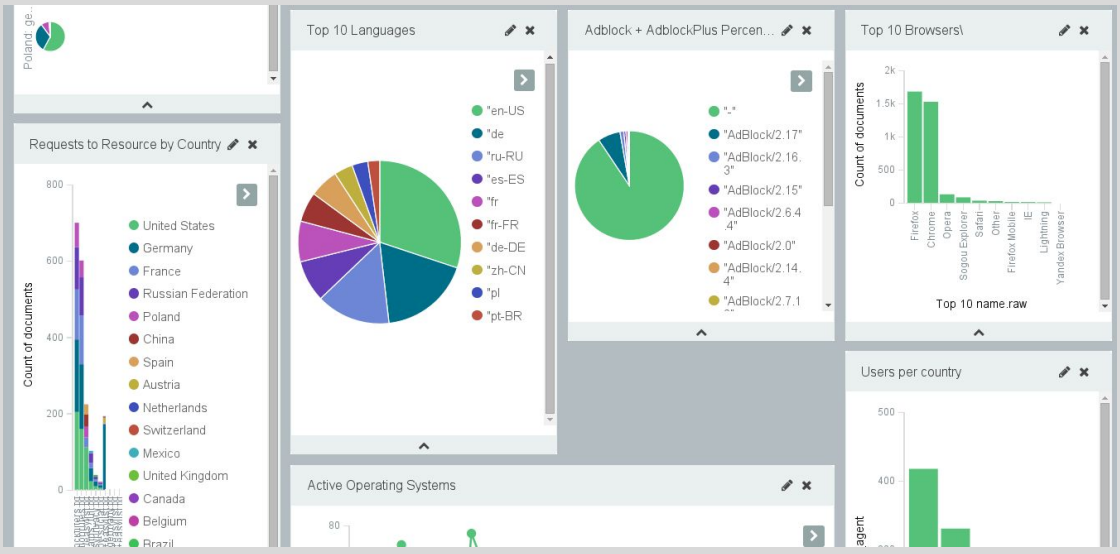
With a live monitoring system, all their questions can be answered in the same moment they ask (or just wait until the processing is done) and manipulate data and graphics to answer their questions as best as possible, converting a file full of data like this:

```
172.56.7.246 - - [17/Nov/2014:00:00:02 +0000] "GET
/antiadblockfilters.txt?addonName=adblockplus&addonVersion=2.6.6&application=fennec2&applicationVersion=33.0&platform=gecko&platformVersion=33.0&
lastVersion=201411151930 HTTP/1.1" 200 1837 "-" "Mozilla/5.0 (Android; Tablet; rv:33.0) Gecko/33.0 Firefox/33.0" "-" https "en-US,en;q=0.5"
"easylist-downloads.adblockplus.org" "-"
174.25.225.23 - - [17/Nov/2014:00:00:02 +0000] "GET
/exceptionrules.txt?addonName=adblockplus&addonVersion=2.6.6&application=firefox&applicationVersion=33.1&platform=gecko&platformVersion=33.1&last
Version=201411152301 HTTP/1.1" 200 41992 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0" "-" https "en-US,en;q=0.5"
"easylist-downloads.adblockplus.org" "-"
115.124.2.226 - - [17/Nov/2014:00:00:02 +0000] "GET /easyprivacy.txt?_=1416182397555 HTTP/1.1" 200 79025 "-" "Mozilla/5.0 (Windows NT 6.3; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.111 Safari/537.36" "-" https "en-GB,en-US;q=0.8,en;q=0.6"
"easylist-downloads.adblockplus.org" "AdBlock/2.6.18"
173.206.224.179 - - [17/Nov/2014:00:00:02 +0000] "GET
/antiadblockfilters.txt?addonName=adblockplus&addonVersion=2.6.6&application=firefox&applicationVersion=33.1&platform=gecko&platformVersion=33.1&la
stVersion=201411160150 HTTP/1.1" 200 1837 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:33.0) Gecko/20100101 Firefox/33.0" "-" https "en-US,en;q=0.5"
"easylist-downloads.adblockplus.org" "-"
123.142.190.92 - - [17/Nov/2014:00:00:02 +0000] "GET
/antiadblockfilters.txt?addonName=adblockplus&addonVersion=2.6.6&application=firefox&applicationVersion=33.0.2&platform=gecko&platformVersion=33.0.
2&lastVersion=201411140450 HTTP/1.1" 200 1837 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0" "-" https
"ko-kr,ko;q=0.8,en-us;q=0.5,en;q=0.3" "easylist-downloads.adblockplus.org" "-"
91.199.196.155 - - [17/Nov/2014:00:00:02 +0000] "GET
/antiadblockfilters.txt?addonName=adblockpluschrome&addonVersion=1.8.7&application=chrome&applicationVersion=38.0.2125.111&platform=chromium&pl
atformVersion=38.0.2125.111&lastVersion=201411151850 HTTP/1.1" 200 1837 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/38.0.2125.111 Safari/537.36" "-" https "ru-RU,ru;q=0.8,en-US;q=0.6,en;q=0.4" "easylist-downloads.adblockplus.org" "-"
108.25.69.21 - - [17/Nov/2014:00:00:02 +0000] "GET
/antiadblockfilters.txt?addonName=adblockpluschrome&addonVersion=1.8.7&application=chrome&applicationVersion=38.0.2125.111&platform=chromium&pl
atformVersion=38.0.2125.111&lastVersion=201411152310 HTTP/1.1" 200 1837 "-" "Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/38.0.2125.111 Safari/537.36" "-" https "en-US,en;q=0.8" "easylist-downloads.adblockplus.org" "-"
125.45.87.86 - - [17/Nov/2014:00:00:02 +0000] "GET
/exceptionrules.txt?addonName=adblockpluschrome&addonVersion=1.6.1&application=chrome&applicationVersion=33.0.1750.146&platform=chromium&platf
ormVersion=33.0.1750.146&lastVersion=0 HTTP/1.1" 200 41876 "-" "Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/33.0.1750.146 BIDUBrowser/6.x Safari/537.36" "-" https "zh-CN,zh;q=0.8" "easylist-downloads.adblockplus.org" "-"
188.135.135.148 - - [17/Nov/2014:00:00:02 +0000] "GET
/antiadblockfilters.txt?addonName=adblockpluschrome&addonVersion=1.8.7&application=chrome&applicationVersion=38.0.2125.111&platform=chromium&pl
atformVersion=38.0.2125.111&lastVersion=201411130840 HTTP/1.1" 200 1831 "-" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/38.0.2125.111 Safari/537.36" "-" https "it-IT,it;q=0.8,en-US;q=0.6,en;q=0.4" "easylist-downloads.adblockplus.org" "-"
118.172.219.249 - - [17/Nov/2014:00:00:02 +0000] "GET
/exceptionrules.txt?addonName=adblockpluschrome&addonVersion=1.7.4&application=chrome&applicationVersion=33.0.1750.146&platform=chromium&platf
ormVersion=33.0.1750.146&lastVersion=201411151951 HTTP/1.1" 200 41876 "-" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/33.0.1750.146 SparkSafe/2.x Safari/537.36" "-" https "en-US;q=0.8,en-US;q=0.6,en;q=0.4" "easylist-downloads.adblockplus.org" "-"
```

[Code 4. Log lines]

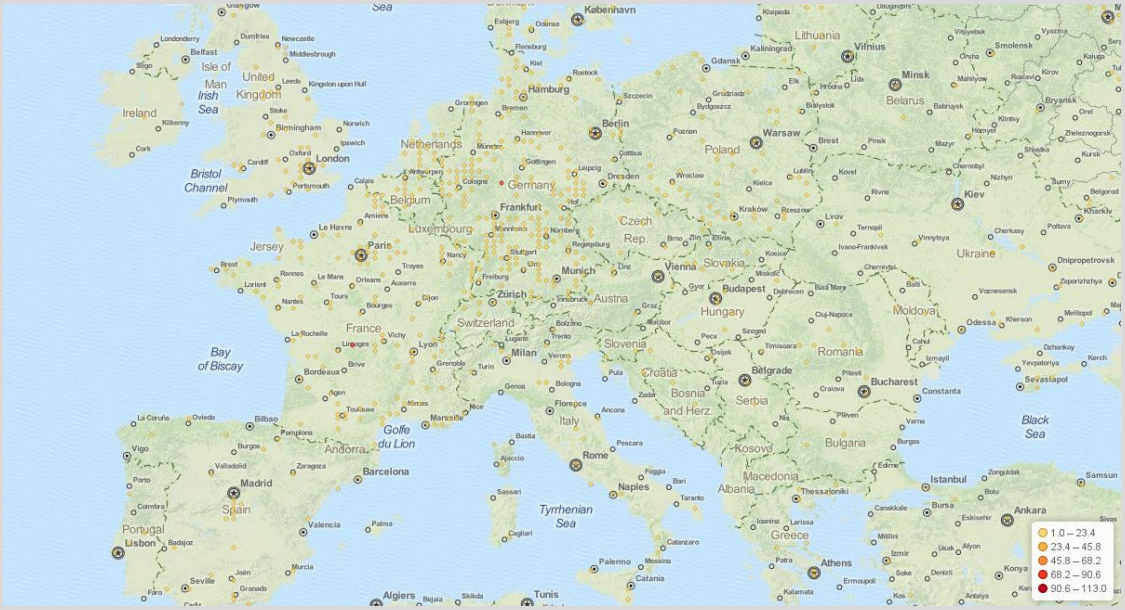
into information like this:

[Figure 7. User Growth Dashboard by Browser and Country]



[Figure 8. Developers Dashboard - Requests by country and browser. Top used languages]

[Figure 9. User Growth Dashboard - Most active Operating systems. Users per Country]



[Figure 10. ABP stats dashboard - Number of requests in Map]

5.5 Reactions of the Team at ABP

People feel attracted with the user interface that Kibana brings to them, and they feel curious since the first moment to try new aggregations and discover new information.

In the same way, developers and non developers realize that they can get access to the information they want and manipulate it in different ways and combine it with their processes to deliver knowledge in a different way, depending on the area of the company.

Having such insight in the information that flows in the background of your servers also gives the possibility to find things that one was not even looking for. Such as:

5.5.1 Insights. Strange details found:

- Possible DDOS Attacks from time to time.
    - This is possible because with the solution the strange behaviour can be seen in the same moment that is happening. With current solution everything is analyzed one day after (best case scenario) and there is no strange behaviour algorithm,
- Some recurrent ips make a lot of requests.
    - Also, with the strange behaviour algorithm that can be selected in Kibana, the IPs that repeat the most can be found easily. With the current solution, a Python or R script has to be programmed and executed to find anomalies.
- Requests are being made to chrome_supplement.txt (apparently it was removed months ago)
    - It looks like no one has even noticed, because there is no need of looking into this specific resource, nevertheless using the aggregations interface that Kibana provides this can be found without troubles.

5.6 Caveats

The company still needs to needs to be trained to use this solution properly as well as put everything to the test to realize if a solution like this fits their needs perfectly.

The deployment of the elasticsearch cluster must be made with care, otherwise only one node running elasticsearch will probably fail and it will not deliver all the benefits that should come with the solution, such as redundancy and failure tolerance.

The first live try with only one node on elasticsearch gathering the data of around 20 servers resulted in a catastrophic crash of the server losing the aggregations made and blocking the data access. This has to be a gradual implementation.

Chapter 6. Discussion

In the previous chapters we chronicled how we implemented an automatic monitoring system, based in an ELK Solution, to facilitate data analysis and scalability. The proposed solution has benefits and disadvantages, all of them became evident once the prototype was implemented.

6.1 The Results

After months of developing and implementing the solution, and while doing it, benefits and challenges were found as the system was growing. This prototype was implemented as an exploratory phase of what could be done to manage and analyze the data. A different amount of observations was gathered. Everything is listed below.

6.1.2 Benefits

The benefits that can be seized out of this solution cover the necessities of the predicted future. With a solution made to scale, not doing so would be like “a Superman that never leaves it’s Clark Kent disguise” ie. (wasted power).

While running the prototype of the ELK stack solution, the following specific benefits were found:

Table 19. Benefits

What?	How?
Insight	allowing real time observation of the data.
Transparency	allowing data to be accessible in different ways
Trustability	because of the load balanced solution of Elasticsearch.
Accuracy in Interpretation	having specific answers to specific questions.
Feedback	is received faster and therefore the changes to improve come faster.
Consistency	processing all the log files in the same way.

Centralization of logs	it's a benefit defining rules to collect and process them in a defined location.
Faster Reaction	with the view and notifications provided with live monitoring.
Defined process	Collect, Parse, Store. A structured process to follow and have a better control in the development and results.
Continuous Improvement	thanks to the scalability and flexibility that the solution extend
Privacy	information is gathered and anonymized such as precise location of IP addresses and yet providing useful information.

And it would **facilitate**:

- **Statistics** thanks to the simplified syntax and flexibility of the system
- **Debugging** keeping track in detail the status of the servers.
- Data **Access** & **Manipulation** using the JSON format that the data processing returns.
- **Decisions** in Business and Development answering to the questions they have in the same moment with updated information.

6.1.3 Limitations and challenges

In ABP user's privacy is a top priority and nothing is ever decided without looking considering it. This makes the gathering of information a more complicated task because there is as less data as possible related to the users, and even so, they need to deliver a better product, a product that behaves as the users expect.

The same way not a lot of data is collected, not a lot of employees in the company have the required privileges to access it. This is a two edged sword; Only one person knowing and having access to the data. Leaving the rest of the company in the dark. They couldn't even answer simple questions like:"How much people is acting like this" or "What kind of message is better for this part of the world to transmit this idea?".

Table 20. Limitations

Data interpretation	It's complicated due to the amount of data stored in the servers.
Information access	Not everyone in the company has the same privileges for every information
Elasticsearch configuration	A lot of new information needs to be learned to establish the appropriate cluster
Comparison with the current solution	There are pros and cons against the current solution, that's why the solution needs to be tested.

First of all, ABP is the most downloaded extension in the world. Even if they don't want to collect a lot of data, they will. Besides, they are just beginning with the real hard work after the deals made with Google and Amazon. The road to build a better navigation experience is giving biggers steps each day. That's why keeping in mind that they will continue growing will happen and they need to know how to handle big amounts of data keeping of course best practices to it's maximum.

6.1.4 The Unexpected Findings

In some experiments, the results of the information that can be obtained with the data was surprising for a lot of people using sentences like "the data analyst said this was not possible", more specific when the Logstash GeoIP location was used to discover where are the requests coming from at what moment of the day.

The last example shows how much there is still to learn about the whole solution. Out of these results, new predictions about the users can be made based on their location, but this information needed to be anonymized to avoid the specific tracking of any user. Here is when the *anonymize* function provided by Logstash was implemented. This way, the balance between information and privacy remains.

There is just too much information about Elasticsearch in the page, no matter how well documented it is, to deal with a lot of information takes longer and can be confusing sometimes. Learn Elasticsearch is a full time task and the number of nodes and the management of its cluster should be taken with special attention. There is where all the data will be anyway.

Thinking of the first part of the architecture, Logstash, it was impressive how the Logstash team really try to facilitate the way the logs are parsed and used for different kind of purposes. The documentation and examples get better each day but of course there are details to improve, for instance loop in a split of a split was not possible in the Logstash versions previous to 1.4.2.

6.2 Observations

6.2.1 About the architecture

6.2.1.1 Logstash

Logstash is very easy to learn and to implement. Thanks to their documentation, all the examples that someone may need are there. And even if the examples are not enough, the high support they give in the IRC channel will help out with any kind of issue.

6.2.1.2 Elasticsearch

Elasticsearch is mainly a search engine and aggregations are made using Javascript, taking longer than other analysis tools. That's why the structure and indexing of the information must be thinked previously and a process to do so must be established before doing it.

Elasticsearch is probably the most complicated part of all the solution. The first deploy appeared to be a success until it crashed mercyless a few minutes later. Then we realized that one Elasticsearch node is not enough for the data to be stored and aggregated in it.

Even while Elasticsearch is a main part of the prototype, different methods could be apply to substitute the way the data is stored, or combine it with the current solutions. This is possible thanks to the flexibility that Logstash offers with the

different output methods such as using technologies like MongoDB, Nagios, Redis and more.

6.2.1.3 Kibana

Kibana is a very well developed tool giving less and less troubles with each released version. The first try was with kibana 3, which is very powerfull (probably still more than the current version 4), but kibana 4 is by far more user friendly. While developing the experiment of this work, Kibana 4 Beta was used almost all the developing time, implementing the stable version the same day it was released.

Working at the sharpest edge with the released technology brings satisfactions and problems when you don't have at your disposal what you expect always testing the last version of the used product.

6.2.1.4 Puppet

One of the advantages of the ELK solution is that it was developed using puppet from the very beginning. Which made it:

- Easily replicable. It can work on any system.
- Self documented and explained

With Puppet, there is no need to worry the worst will happen if someone touches the wrong button inside the server. The idempotency[8] that puppet brings to the game is a key advantage.

6.2.1.5 ELK Stack

In my opinion an ELK Solution is recommended for any company who wants to keep things simple and will grow exponentially in their Log Processing needs. Absolutely worth trying.

[8] Is the property of certain operations in mathematics and computer science, that can be applied multiple times without changing the result beyond the initial application.

An important thing to mention about the ELK stack is how active they are on IRC and the community is always very helpful. If the documentation is not enough, someone will always reply on IRC, or even looking deep into the code if something strange happens, These are some examples of the magic that can be achieved in Open Source projects.

To implement this ELK solution the best would be to have previous experience with Elasticsearch or at least be prepared to deal with a lot of information and learn a lot out of it. Reliability tips and architectures must be studied before (all documented in the Elasticsearch guide) along with deploy information and maintenance of the clusters.

Before making all the aggregation and deep analysis of the data, a plan should be made to evaluate what kind of aggregation can be made in which part of the architecture. Data can be aggregated using Logstash, Elasticsearch or even Kibana. This analysis will improve performance and will make the solution more understandable obtaining the expected results.

6.3 Summary

One of the observation of the Data Analysis person in ABP was that with the current solution it took 20 mins. to process all the data of the day... How long would it take wit the ELK Solution? We had to explain that it was in real time, so processing was continuous... and the question didn't apply anymore.

In ABP an open culture can be seen from the distance. They are always thinking the best way to make a better company and a better product. That’s how the development of this solution became easier, with the help of the different areas and specially the team work with the analysis and the infrastructure areas.

In the beginning we had the following questions in the back of our head.

- If there is not much data, why is the ELK stack needed?
    - Scalability
    - Ease of coding and output
    - Better organization of data
    - Due to the amount of users, they produce a lot of data
- How can the data be analyzed deeply without interfering in the user privacy?

What we have is good, but still not perfect. As discussed before, there are a lot of areas to grow and to take the best out of these tools. At ABP the ELK stack will be put on test and only the time will tell all the final conclusions.

7. Conclusions

Information is power. And,*"with great power comes great responsibility"*. AdBlock Plus is aware of this and tries to find the ethical balance between the information they collect to improve as a company and to respect their user's privacy. They take care of the their 60 million active users by gathering the least amount of data possible while still having enough information to work and make decisions.

The analysis of immense amounts of log files is a challenge. Currently, most companies would use a combination of custom made scripts, tools and Excel files to collect, analyse and show this data. In ABP the traditional approach has the following disadvantages:

- Data is analyzed at most once a day
- Decisions are taken at most once a week
- Data analysis was centralized in one person
- Custom inquiries by other departments, took the data analyst at least a week to be solved.

As an exploratory prototype, a new way to dig into their data was implemented: Logstash, Elasticsearch and Kibana (the ELK Solution). Three open source tools that combined give better insight to the company and facilitates the log analysis tasks, were implemented to produce a lot of information out of their generated data.

The ELK Solution gave benefits such as:

- Centralization of analysis, removing complexity while parsing and cleaning the logs
- Continuous processing of the data provides information in almost real time which leads to better decision making.
- Ability to answer custom inquiries by other departments in minutes instead of weeks.
- Faster deletion of logs and anonymization of sensitive fields which protects user's privacy.

- A centralized and consistent dashboard for decision makers.
- Horizontally scalable solution that will grow as ABP grows.

Along with the develop of the prototyped solution, lessons of implementation and usage were learned and the document shows how this is a viable solution for companies that still doesn't have a defined process to analyze the information they gather but extra precautions need to be taken into account when establishing the Elasticsearch cluster. This has to be carefully evaluated to decide the amount of necessary nodes to have. The documentation can and must be followed to improve areas like reliability and performance, nevertheless the amount of documentation that they provide could lead to a time risk: it is too much and a lot of details need to be considered.

The full automated prototype became easy to maintain, high scalable and reliable. It will be put to the test now in production to learn more of it.

So far the ELK Solution has proven to be an easier approach to analyse and parse the generated data of the company and if power is what is being sought, this is an excellent path.

References

Adblockplus.org,. (2015). Adblock Plus - Surf the web without annoying ads!. Retrieved 25 April 2015, from https://adblockplus.org

Adweek.com (2013). Social Times - How Facebook Handles 300 Petabytes Of Daily Data. Retrieved 29 April 2015, from http://www.adweek.com/socialtimes/presto/429910

Basin, D., Schaller, P., & Schläpfer, M. (2011). Applied information security. Berlin: Springer.

Chuvakin, A., Schmidt, K., Phillips, C., & Moulder, P. (2013). Logging and log management. Waltham, Mass.: Syngress.

Cid, D. (2006). Log analysis for intrusion detection. (p. 1).

Cohen, D. (2013). How Facebook Handles 300 Petabytes Of Daily Data. Adweek.com. Retrieved 25 April 2015, from http://www.adweek.com/socialtimes/presto/429910

Docs.oracle.com,. (2015). What Is Data Mining?. Retrieved 5 May 2015, from http://docs.oracle.com/cd/B28359_01/datamine.111/b28129/process.htm#DMCON002

Dean, J., & Ghemawat, S. (2008). MapReduce. Commun. ACM, 51(1), 107. doi:10.1145/1327452.1327492

Duarte, O. (2014). Explorando Big Data a través de ejercicios prácticos. Mexico: Centro de Investigación en Matemáticas AC.

Traverso, M,. (2015). Presto: Interacting with petabytes of data at Facebook. Retrieved 27 April 2015, from https://www.facebook.com/notes/facebook-engineering/presto-interacting-with-petabytes-of-data-at-facebook/10151786197628920

Harris, D. (2015). Airbnb open sources SQL tool built on Facebook's Presto database. Gigaom.com. Retrieved 25 April 2015, from https://gigaom.com/2015/03/05/airbnb-open-sources-sql-tool-built-on-facebooks-presto-database/

Hive.apache.org,. (2015). Apache Hive TM. Retrieved 27 April 2015, from https://hive.apache.org/

Jansen, Bernard J. "Search log analysis: What it is, what's been done, how to do it." Library & information science research 28.3 (2006)

Kennedy, J. (1983). Analyzing qualitative data. New York, N.Y.: Praeger.

Mayer-Schönberger, V., & Cukier, K. (2013). Big data. Boston: Houghton Mifflin Harcourt.

Melnik, S., Gubarev, A., Long, J., Romer, G., Shivakumar, S., Tolton, M., & Vassilakis, T. (2011). Dremel.Commun. ACM, 54(6), 114. doi:10.1145/1953122.1953148

Miller, R., Willis, R., & Interactive, B. (2012). Estimate: Amazon Cloud Backed by 450,000 Servers | Data Center Knowledge. Data Center Knowledge. Retrieved 25 April 2015, from http://www.datacenterknowledge.com/archives/2012/03/14/estimate-amazon-cloud-backed-by-450000-servers/

Nagios.org,. (2015). Nagios - About Nagios. Retrieved 5 May 2015, from http://www.nagios.org/about

Olindata.com,. (2015). What is Puppet? Why use Puppet?. Retrieved 6 May 2015, from http://www.olindata.com/technology/puppet

Oliner, A., Ganapathi, A., & Xu, W. (2011). Advances and Challenges in Log Analysis - ACM Queue. Queue.acm.org. Retrieved 27 April 2015, from https://queue.acm.org/detail.cfm?id=2082137

Orzell, G. (2012). The Netflix Tech Blog: Announcing Servo. Techblog.netflix.com. Retrieved 25 April 2015, from http://techblog.netflix.com/2012/02/announcing-servo.html

Prestodb.io,. (2015). Presto | Distributed SQL Query Engine for Big Data. Retrieved 25 April 2015, from https://prestodb.io/

Puppet Labs,. (2015). What is Puppet?. Retrieved 6 May 2015, from https://puppetlabs.com/puppet/what-is-puppet

Sissel, J (2012). Logging: logstash and other things. Puppet Conf 2012. Retrieved from: https://www.youtube.com/watch?v=RuUFnog29M4

Solano, J., & Leiva, E. (2014). Big Data Analytics: propuesta de una arquitectura (p. 1). Costa Rica: Universidad Latinoamericana de Ciencia y Tecnología,.

Spark.apache.org,. (2015). Apache Spark™ - Lightning-Fast Cluster Computing. Retrieved 25 April 2015, from https://spark.apache.org/

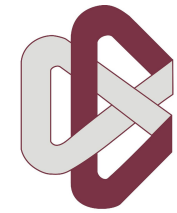
StorageServers,. (2013). Facts and Stats of World's largest data centers. Retrieved 26 April 2015, from https://storageservers.wordpress.com/2013/07/17/facts-and-stats-of-worlds-largest-data-centers/

Sumo Logic,. (2015). Netflix Selects Sumo Logic's Next-Gen Log Management and Analytics Service - Sumo Logic. Retrieved 25 April 2015, from https://www.sumologic.com/news/2012-09-11/netflix-selects-sumo-logics-next-gen-log-management-and-analytics-service/

Turnbull, J. (2013). The logstash book. Kindle Edition.

Vaarandi, R. (2005). Tools and Techniques for Event Log Analysis (p. 1). Estonia: TALLINN UNIVERSITY OF TECHNOLOGY.

Zhang, Qi, Lu Cheng, and Raouf Boutaba. 'Cloud Computing: State-Of-The-Art And Research Challenges'. J Internet Serv Appl 1.1 (2010): 7-18. Web.



CIMAT

CENTRO DE INVESTIGACIÓN EN MATEMÁTICAS, A.C.

BIBLIOTECA

AUTORIZACION PUBLICACION EN FORMATO ELECTRÓNICO DE TESIS

El que suscribe

Autor(s) de la tesis: Alfonso Alvarez Sanchez.

Título de la tesis: Live Big Data Logs Analysis.

Institución y Lugar: Zacatecas, Mexico

Grado Académico: Licenciatura ( ) Maestría (X) Doctorado ( ) Otro ( )

Año de presentación: 2015

Área de Especialidad: Software

Director(es) de Tesis: José G. Hernandez

Correo electrónico: alfonso.alvz@gmail.com

Domicilio: Priv. de Querétaro 109

Palabra(s) Clave(s): Big Data, Log Analysis, Realtime, Logstash, Elasticsearch, Kibana, Automatic Provisioning, Open Source

---

Por medio del presente documento autorizo en forma gratuita a que la Tesis arriba citada sea divulgada y reproducida para publicarla mediante almacenamiento electrónico que permita acceso al público a leerla y conocerla visualmente, así como a comunicarla públicamente en la Página WEB del CIMAT.

La vigencia de la presente autorización es por un periodo de 3 años a partir de la firma de presente instrumento, quedando en el entendido de que dicho plazo podrá prorrogar automáticamente por periodos iguales, si durante dicho tiempo no se revoca la autorización por escrito con acuse de recibo de parte de alguna autoridad del CIMAT.

La única contraprestación que condiciona la presente autorización es la del reconocimiento del nombre del autor en la publicación que se haga de la misma.

Atentamente

_____

Nombre y firma del tesista

CALLE JALISCO S/N MINERAL DE VALENCIANA APDO. POSTAL 402 C.P. 36240
GUANAJUATO, GTO., MÉXICO TELÉFONOS (473) 732 7155, (473) 735 0800 EXT. 49609 FAX.
(473) 732 5749 E-mail: biblioteca@cimat.mx