

A DATA ANALYTIC MODULE TO EXTEND GRAFANA FUNCTIONALITY

BY

GAN KIAN YONG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Kampar Campus)

JANUARY 2019

REPORT STATUS DECLARATION FORM

Title: A Data Analytic Module To Extend Grafana Functionality

Academic Session: January 2018

I GAN KIAN YONG
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:

NO 62 JALAN 2, JALAN KELANG

RUMAH MURAH

45000 KUALA SELANGOR

SELANGOR

(Supervisor's Name)

Date: _____

Date: _____

A DATA ANALYTIC MODULE TO EXTEND GRAFANA FUNCTIONALITY

BY

GAN KIAN YONG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Kampar Campus)

JANUARY 2019

DECLARATION OF ORIGINALITY

I declare that this report entitled “**A DATA ANALYTIC MODULE TO EXTEND GRAFANA FUNCTIONALITY**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : GAN KIAN YONG

Date :

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my project supervisor, Dr Ooi Boon Yaik who has taught and guided me throughout the development of final year project. His insight and teaching method played a great influence on guiding me to solve problems.

Besides, I would like to thank to the CEO of Exabytes, Chan Kee Siak who sponsor the cluster and data in elasticsearch for this project, help me to conduct my final year project.

Last but not least, I would like to thank to CREST (Collaborative Research In Engineering, Science and Technology) sponsor to this project, they support me on develop the final year project.

ABSTRACT

In this data driven era and the concept of industry 4.0, they will need the versatile platforms to visualize and analyse their event data to explore, interpret and understand the information hide in the data. A versatile data analytic platform can help to improve and assist the various kind of analytics. Those analytics including the descriptive, predictive and the advance prescriptive analytics. The Data analysis is an important process that make data become more valuable and gain more insights about the data. Various tools and systems used to analyse the huge volume of complex data to gain the insight from the raw data which is meaningless. In this project, we compare the strengths and weaknesses of the solutions, approaches and the popular visualization tools done by others. The candidate including the Amazon (Amazon Kinesis and Amazon QuickSights), Microsoft Azure (Time Series Insights), IBM (Watson), InfluxDB, Elasticsearch, OpenTSDB, Grafana and Kibana. We say that Grafana outrank the rest in terms of the dashboard features, the pricing, functionality and flexibility. Elasticsearch have a powerful search engine, it has a better query throughput which is suitable to handle large amount of data. We solve the limitation of Grafana that cannot perform custom query and visualize data from the custom query. We propose a solution to enhance the data analytic platform and improving the system extension functionality by develop a programme that sit between the data source (Elasticsearch) and the visualization tool Grafana. The new or advance analytics done by the data scientists can be added into the system through this program in the future easily.

TABLE OF CONTENTS

TITLE PAGE	i
DECLARATION OF ORIGINALITY	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF CODE SEGMENTS	x
LIST OF ABBREVIATIONS	xi
Chapter 1 Introduction	1
1.1 Project Background	1
1.2 Problem Statement and Motivation	2
1.3 Objectives	4
1.4 Proposed Approach	5
1.5 Impact, Significance and Contribution	6
1.6 Organization of the Report	7
Chapter 2 Literature Review	9
2.1 Amazon (Amazon Kinesis and Amazon QuickSights)	10
2.2 Microsoft Azure (Time Series Insights, TSI)	13
2.3 IBM (Watson)	15
2.4 InfluxDB	17
2.5 Graphite	17
2.6 OpenTSDB	17
2.7 Elasticsearch	18
2.8 Grafana	18
2.9 Kibana	18

2.10 Limitation and weakness of the cloud service method	19
2.11 Comparison between Graphite, InfluxDB, OpenTSDB and Elasticsearch	20
2.12 Comparison between Grafana and Kibana	22
Chapter 3 System Design	24
3.1 Use case design	24
3.2 Use Case Description	25
3.3 Activity Diagram.....	30
3.4 Sequence Diagram	31
3.5 Communication Diagram	31
3.6 Timeline.....	32
Chapter 4 System Implementation	33
4.1 Methodologies	33
4.2 Tools to use.....	34
4.3 Processing data for visualization	35
4.4 Visualize data from advance query function	38
4.5 Implementation issue and Challenge	42
Chapter 5 Experimental Result	43
5.1 Analysis Function and data souce support by Grafana.....	43
5.2 Visualization function of Grafana	46
5.3 Extensibility of Grafana	47
Chapter 6 Conclusion.....	49
Bibliography	51
Appendix A Poster	58
Appendix B Turnitin Result	59
Appendix C Supervisor’s Comments on Originality Report.....	61
Appendix D Checklist for FYP2 Thesis Submission	62

LIST OF TABLES

Table Number	Title	Page
Table 2- 1	Comparison between Graphite, InfluxDB, OpenTSDB and Elasticsearch	20
Table 3- 1	Refresh the dashboard use case description	25
Table 3- 2	Enter request parameter use case description	26
Table 3- 3	Set HTTP request path use case description	27
Table 3- 4	View the display output use case description	28
Table 3- 5	add extension service to system use case description	29

LIST OF FIGURES

Figure Number	Title	Page
Figure 1- 1	System Architecture	5
Figure 2- 1	Amazon Kinesis and Amazon QuickSights Architecture.	10
Figure 2- 2	Propel your IoT platform to the cloud with Azure Time Series Insight	13
Figure 2- 3	IBM IoT overview.	15
Figure 2- 4	Comparison of write performance for InfluxDB and Elasticsearch	21
Figure 2- 5	Comparison of the query throughput for InfluxDB and Elasticsearch	21
Figure 3- 1	Use case design of Grafana Data Analytic System	24
Figure 3- 2	Activity Diagram Grafana analytic module	30
Figure 3- 3	Sequence Diagram of Grafana Analytic Module	31
Figure 3- 4	Communication diagram of Grafana analytic module	31
Figure 3- 5	Gantt Chart of FYP1 and FYP2 Gantt Chart of FYP1 and FYP2	32
Figure 4- 1	A prototyping-based methodology	
Figure 4- 2	Different panels available on the Grafana	35
Figure 4- 3	Available option for each panel	35
Figure 4- 4	Panel in the edit mode	35
Figure 4- 5	Modified query request in the panel	36
Figure 4- 6	Show the data display to Grafana into JSON format	37
Figure 4- 7	Data visualization on the panel from the JSON string	38
Figure 4- 8	AJAX panel on grafana	38
Figure 4- 9	Show the environment in the edit mode of the AJAX panel	39
Figure 4- 10	HTTP request sent from Grafana (AJAX)	39
Figure 4- 11	Result of the AJAX panel	41
Figure 4- 12	Different panels display on the Grafana	41
Figure 5- 1	Data sources supported by primitive Grafana analytic platform	43
Figure 5- 2	Analysis function available in the panel from primitive Grafana analytic platform	44
Figure 5- 3	Query used in the improved Grafana analytic platform	44

Figure 5- 4	The name of different data source to handle query request	45
Figure 5- 5	The name of the analysis file in the server	45
Figure 5- 6	Different kind of panels in the dashboard	46
Figure 5- 7	Bar chart in the improved Grafana analytic platform	46
Figure 5- 8	Show the pie chart in the improved Grafana analytic platform	47
Figure 5- 9	Function available in the primitive Grafana analytic platform	47
Figure 5- 10	Advance analysis function for the improved Grafana analytic platform	48

LIST OF CODE SEGMENTS

Code Segment	Title	Page
Code Segment 4- 1	Process the incoming query request	36
Code Segment 4- 2	Get data from the analysis function	36
Code Segment 4- 3	Show the dataframe to the Grafana	37
Code Segment 4- 4	Store the parameters from the HTTP request	40
Code Segment 4- 5	Determine the analysis file require to be execute and display the images	40

LIST OF ABBREVIATIONS

API	Application Programming Interface
BI	Business Intelligent
DBMS	Database Management System
HTTP	HyperText Transfer Protocol
IBM	International Business Machines
InfluxDB	Influx Database
InfluxQL	Influx Query Language
IoT	Interne Of Things
OpenTSDB	Open Time Series Database
RAD	Rapid Application Development
SSH	Secure Shell
SQL	Standard Query Language
TSI	Time Series Insights
UI	User Interface
JSON	JavaScript Object Notation

Chapter 1 Introduction

1.1 Project Background

Internet of things analytic is an important application of data analysis and a procedures to explore a huge volume of raw data which collect and capture from those IoT sensor continuously. The cloud end of the IoT consist of three important segment, which include data visualization, storage and the analytic platform. All the raw data capture from the sensor are meaningless and valueless for everyone, it require some analysis and visualization tool to interpret in order to understand the information hiding in the data.

Data visualization is a term that describing the ability to help people understand and explain the large amount of complex data by putting it in a graph, charts or any visual context. (Pittenturf n.d.) Data visualization place an important role in data analysis as it make complex data more understandable and usable. People can understand the information form the data easier and comprehend the tendency by looking at the trend pattern from the graph. Visualize and interpret the complex data help business leader understand information and provide the ability easily drill drop into data identify their weakness in order to improve their business grow. Today's data visualization has become standard for recent business intelligence to support better business decision making and get significant competitive advantage over their business competitor. (Rouse, 2017)

Data storage is describing recording data into a storage medium, ensuring access and update managed of data easily and effectively. IoT sensor device always have limited storage capability, time-base data capture from the IoT sensor device will send to the cloud however high volume of data that stored in the cloud will slow down the process of accessing the data. (Gerber, 2017) Time series databases is build and design specifically for time-base data storage, it has high availability, support concurrent read and write process and allow high transaction volume of time series

data, query performance can be improve by configure the index to optimize data access.

Analytic platform help to perform operation on data analytic, it is a solution design to meet the demand of users or businessman who have large data-drive companies. It provide detail analysis in contextual form from all the data stored though descriptive, predictive and prescriptive analytic. Descriptive analytic to gain insight into the past where it allow us to observe what behaviours happened from the past and understand how it will affect the result and outcome in the future while the predictive analytic make use of some statistical model and various algorithm to identify the relationship among different data set to predict things might happen in future. From the predictive analytics, it provide companies with the actionable insights base on the estimated likelihood and probabilities of a future outcome.

Prescriptive Analytic will provide some recommendation, advice or some actions to guild the companies solve a problem, take benefit from the prediction. (Rijmenam, 2017)

1.2 Problem Statement and Motivation

Grafana is a free open-source, feature rich, analytic and monitoring software which run as a web application to compose dashboard and graph. Grafana is the one of the popular visualization and monitoring tools, it supports more than 30 open source and various type of data sources that store time series data. (Kili, 2018)

InfluxDB is one of the backend, an open source Time Series database that have high-availability to storage the time series data especially the data from IoT sensor and data to be analyze in real time. Elasticsearch is another backend of Grafana, an open source in Java-based and it is built on top of the Apache Lucene. Elasticsearch can search, store and retrieve index document files in various structure and format in real time. Grafana is a well recommended tool to visualize the data from the InfluxDB and Elasticsearch.

Grafana has a tight and close relationship with those database to perform some analysis, visualization and monitoring operation. However, every graph that visualize

from the Grafana are query to the database, every function are perform by the database and result sent back to the Grafana for visualize and display. This is one major drawback where the function available on the Grafana are highly depend and rely on the function provide from the database. (Yigal, 2018)Some users or organizations facing a problem where some visualization functions needed does not provide by Grafana.

- Grafana cannot perform customize query function
- Grafana cannot visualize certain customize function, because it unable to perform custom query function
- New applications or analysis function develop from those data scientists cannot plugin to the system directly to provide new functions.

The problem is important because some organization require to perform some custom function to gain certain insight and understand the information from the data to support and assist their business grow. Since Grafana cannot solve the custom query function and visualize the data, users may require some data scientists to develop application to conduct analysis and generate report for them. Those data scientists have to generate report again and again for user which is troublesome and not effective. This project would like to develop a program to enhance the existing analytic platform, increase the extension functionality of system and application develop by the data scientists and plugin to the existing system though the program. Custom query function problem able to resolve, data scientists don require to generate report again.

1.3 Objectives

The objective of this project is develop a program that seat at the middle between Grafana and data source to solve the problem and limitation of Grafana. An important function provide by the program is the ability to redirect the query request and further process the custom query by other programs or apps. The new function or apps able to plugin directly through this program and enable Grafana to visualize the custom query function and data. The program have more control on the query request send from one end to another end as all request have to go through the program first before reaching another end, so the program have the ability to modify the query or data before return the result.

. The program also increase the compatibility of the system where the new applications created from those data scientists or other customized extension function able to plug into the system by connecting the application to the program. This function able to solve the problem and limitation of Grafana that cannot perform custom query function, because the query function from the Grafana no longer depend on and limited by the database anymore. The program will not break and affect the original function of Grafana and the database if the program perform nothing on the query, the process and query result will work fine as usual. Once the custom query function able to perform, Grafana able to visualize the data from the custom query function.

This project aim to make the IoT analytic platform easier to be use for everyone. The new application or any extension service created by data scientists can be add into the system easily, need not concern on the database and don require to generate the report repeatedly. While user can use the platform to visualize and analyse their data without any complex configuration and coding. This project will not focus on the scope about IoT data visualization, storage medium and analysis functions but focus on developing a framework to support and enhance the Grafana IoT data analytic platform by improving the system's extension functionality.

1.4 Proposed Approach

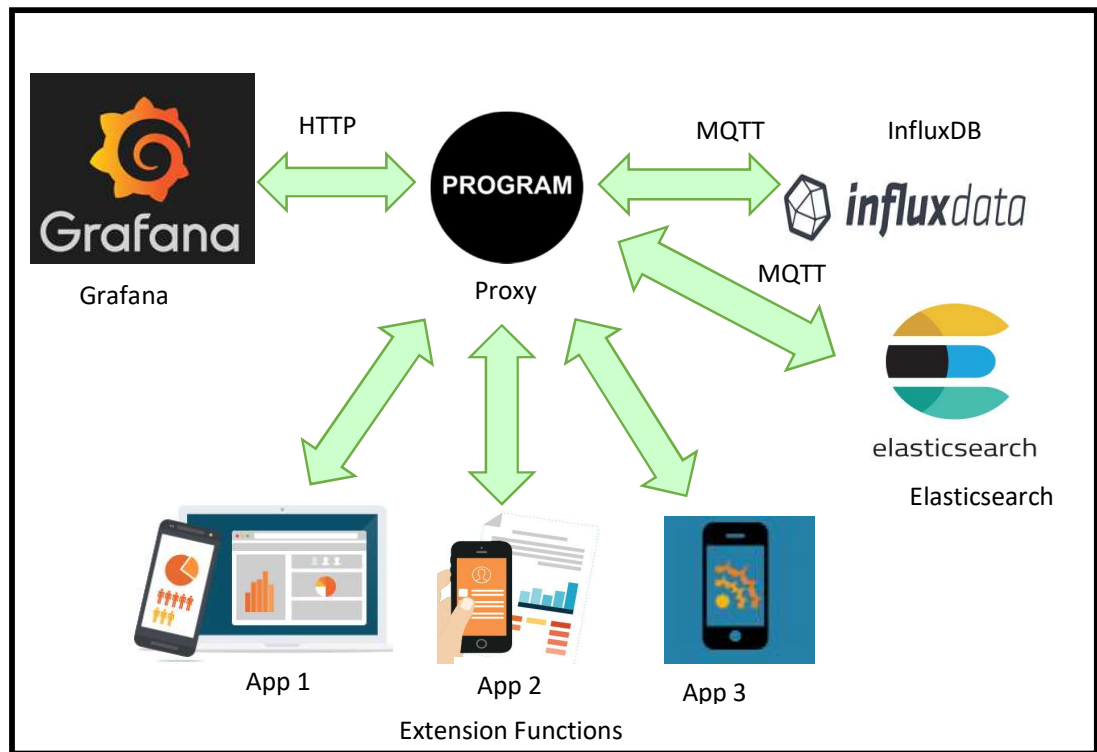


Figure 1- 1 System Architecture

This project going to deliver a program to improve the IoT analytic platform by modifying the current system instead of constructing a new IoT analytic system. The program increase the extension functionality of the system and able to cover and support various type of database. A program is to be created that sit between the data source and the Grafana. The new program or function that able to perform custom function can be plugin to the system through this program, the result will be send back to the Grafana for display.

During this stage, the program will be primarily built and designed for Elasticsearch. However, the program is expected to work well for other data source such as MySQL, InfluxDB and OpenTSDB, with minimal system configuration and modification. The framework provide the system a better analytic platform for user to use minimal configuration and coding, the framework also provide a platform for those data scientists to develop the new application or any advance analysis function into the system easily.

1.5 Impact, Significance and Contribution

A module or framework that able to increase the system's extension functionality is an important keys in the data analytic, where the system able to accommodate other new apps or programs in future. The custom query function is a useful function for any society, business or organization, this module able solve their problem and support their business grow. User may not have knowledge and understand on how the procedure of analysis data work, coding behind the system and the complex configurations, with this project users still able use the system well to visualize their event data. On the other hand, the data scientists can focus on developing their advance analysis applications regardless of the databases use. Once their application plugin to the system though this module, users can perform various query, analysis and visualize different of data in the system independently. This framework make the whole system, Grafana analytic platform easier to be use for everyone. More custom function and advance analysis can be conduct though this platform in future.

From this module, others people or community can explore and enhance the system though different way. They can

- Provide more extension services to the system to conduct more advance or custom analytic.
- Provide system more access control, enhance the system in terms of security perspective.
- Enable system to perform accounting or reporting function
- Enable system to perform load balancing function, increase the performance of system
- Increase system availability and high fault tolerant, system become more robust.

1.6 Organization of the Report

This report consists of 6 chapters, in addition to the appendix and references. The first chapter emphasized and highlighted the description about this project, which include the project background, problem that intended to solve as stated in the problem statement and motivation, objective of this project, proposed approach, and the significant impact and contribution of this project. The second chapter shows the information about literature review and compare the existing solution with the proposed system in terms of the many perspective. Chapter 3 will shows the overview of the proposed system, user requirements of this project. Chapter 4 shows the description about the implementation of the system and the implementation issue. For chapter 5, the experimental result of this project shown and compare the existing solution with the proposed solution. Chapter 6 is the conclusion of this project, summarize of this project including the problem statement and motivation, the future work of this project are stated.

Chapter 2 Literature Review

In current practice, there are some solutions and approaches being used to perform the data analytic on the huge volume of time base data. Basically, those approaches can be category into two type of approaches which is cloud service approaches and on-premise approaches. For the cloud service approach, they are some companies are trying to place themselves as the leader in the Internet of Things revolution with their powerful system. Firstly, the most popular IoT analytic platforms available today will be Amazon Web services IoT platform. Amazon has a very large market share and dominating the consumer in cloud market. Moreover, Microsoft Azure, IBM are the famous company that provide good analytic services and platform in the market today.

Besides, some engine and tool are applied on some time series database to perform the IoT data analytic. The databases such as InfluxDB, Graphite and OpenTSDB are the popular time series database are design specifically to store and deal with the timely data, InfluxDB has win the 2018 DEVIES award from DeveloperWeek for the IoT data time series platform. (Herring, 2018) While the Graphite, Elasticsearch and OpenTSDB are also the top and popular database available today. Grafana and Kibana are the open source visualization and application monitoring tool, three of them are top and well recommended application tools to monitor the IoT time based data. (Acreman, 2016)

Based on the two different type of approach and candidates mentioned above, literature reviews will be done to investigate and identify their approach, strengths and weaknesses of their solution toward IoT data analytic. Compare each and every approach based on some feature and characteristics on their solution.

System Approach

2.1 Amazon (Amazon Kinesis and Amazon QuickSights)

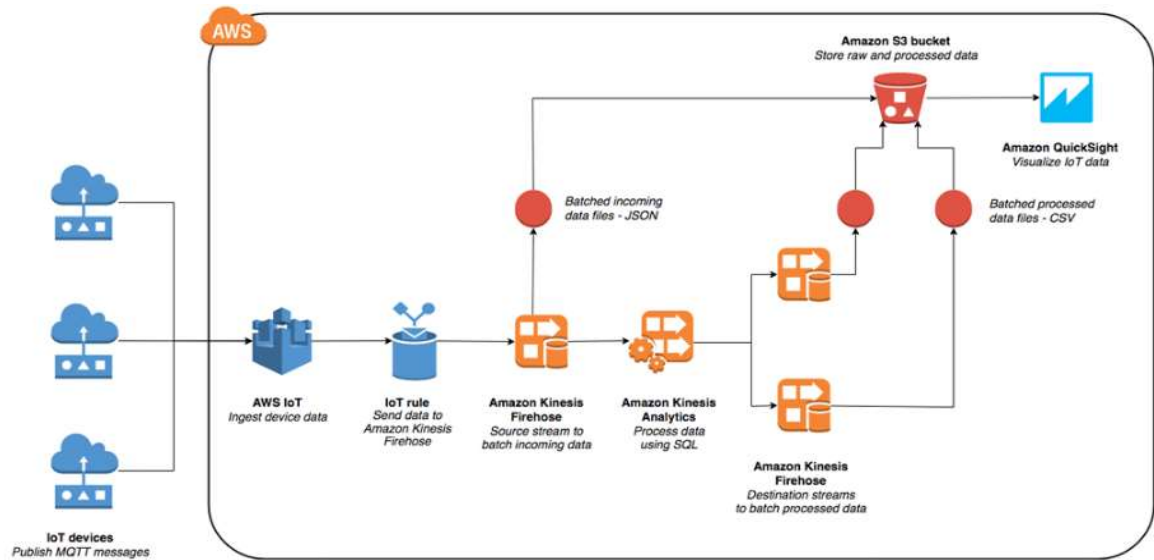


Figure 2- 1 Amazon Kinesis and Amazon QuickSights Architecture. (Desai 2017)

The IoT analytic solution for Amazon involve Amazon Kinesis and Amazon QuickSights. Amazon Kinesis is a service that help to collect, analyze and process the event date to get the timely insight. Kinesis able to capture and store up to terabytes of event data in an hour from various source or event. Kinesis contain (KCL) Kinesis Client Library, users can build their Kinesis application, with the large volume time series data generate a real time dashboard, trigger an alerts once the anomalies detected. Amazon Kinesis allow users make a query to the streaming data using standard SQL to gain the timely insight that support business grow and respond to the customer needs. (Amazon Kinesis n.d.)

Amazon QuickSights is a cloud service that help to visualize the analyze data to get the business insight from the timely data and perform ad-hoc analysis. The graphical tools and rich dashboards provided by the Amazon QuickSights to support the users to build their data visualizations. The data can be pull from various sources including Amazon DynamoDB, Amazon Kinesis, Amazon Relational Database Service and others. (Amazon QuickSights, 2016) It have a good responsive query

performance which boosted by SPICE it stand for (Super-fast, Parallel, In-Memory Calculation Engine). Every month, each user have to pay 9 US Dollar for the standard edition of the Amazon QuickSights. (Heath, 2016).

Strength(s) of Amazon Kinesis and Amazon QuickSights

- Amazon Kinesis enable nearly real-time data processing with a very small latencies. It allow users to continuously interact and monitor the incoming data, analyse and provide the timely insight in seconds, not require analyse until all data being collected. It provide many built-in function to support the advance analysis and detection.
- Kinesis provide high reliability that synchronously duplicates users streaming data that flow into the system and preserves the event data 7 days to prevent data loss due the event or application failure or other machine fault. Amazon QuickSights also require users to pay only for the session they access, no additional charge for not active user or any upfront cost. (Amazon Kinesis n.d.)
- Support standard SQL and easy to use, Amazon Kinesis allow users to build the streaming application and make query using the standard SQL, need not users to learn any complex framework or new programming languages. (Amazon Kinesis Data Analytics n.d.)
- Kinesis is scalable as flow of input data change, it will scales and control the resource and users only require to pay for resources used, not require upfront cost.
- Amazon QuickSights easier to be use and support various data sources, data visualization easier for users to understand the information about the data and provide recommendation the suitable or appropriate type of visualization analysis base on the input data. The visualization able to share will others. Dashboard is snapshot of analysis that allow user to share the view to other users, other users able to edit and filter the dashboard too.

Drawbacks:

- Amazon Kinesis only allow timely data alive within twenty-four hour only, any data that are older will be delete automatically, users have to always ensure that the throughput produce the correct result within that time frame. If users need to work with the data over the twenty-four hour, user can send their data to DynamoDB to process the data. (Flynn, 2014)
- The limitation of the Amazon QuickSights where the table and query result that important into SPICE must be less than the space of SPICE and maximum up to 25GB. Filter maybe use to decrease the data size in order to retrieve data from a larger table. The query on a table maximum up to 1000 columns, while the column name must be lesser than 127 characters.
- Data must be in the format that supported, It support the decimal data type maximum up to four decimal places only, any value that more four decimal place will be truncated. Users have to always make sure that the data can be implicitly convert to the supported data type else QuickSights will tend to skip the rows that unable to convert. (Data Source Limits n.d.)

2.2 Microsoft Azure (Time Series Insights, TSI)

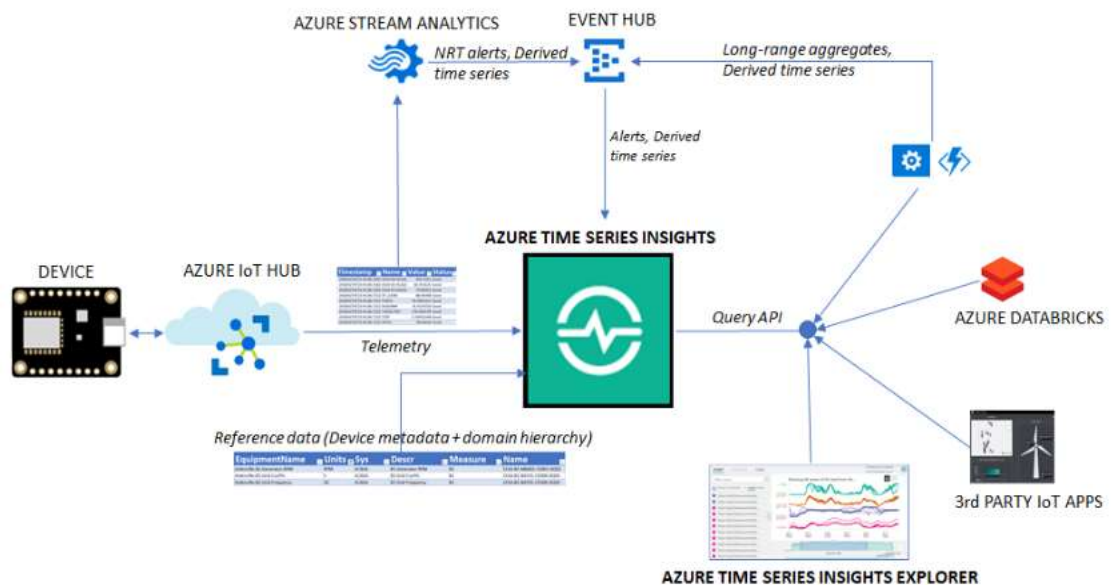


Figure 2- 2 Propel your IoT platform to the cloud with Azure Time Series Insight.

(Shankarnarayan 2018)

The solution for Microsoft Azure to analyze time series data in a fully cloud-based system and involve Azure services. TSI (Time Series Insights) help to analyze a huge volume of time based data in near real time and provide a high capacity of storage to store the data. TSI is a fully managed solution, analytics, visualization and storage service that help to manage IoT time based data in the cloud. It is hosted in the cloud and provide with many useful development tools to analyze the huge volume time based data so it need not require the upfront data preparation by the enterprise and prepare their own IT resources for development. TSI involve in stream processing and analytics nearly real time to explore and correlate various of data sources, companies able to predict future change by looking the information and trend from the previous data, forecasting result by comparing the new data with the previous historical data. TSI provide tools to visualize the trends and alerting services to detect the anomalies. (Azure Time Series Insights n.d.)

Strengths of Time Series Insight

- TSI have high scalability to store the time series data, it have a database design for time based data and support IoT at scale. It allow one million to hundred million of event or data flow in a day and provide full manage and service to the events.
- The view of the time series data streaming able to share to multiple site and locations. The Time Series Insight environment can be connected from different sources and different location to view the streaming data together simultaneously. (Azure Time Series Insights, 2018)
- The shared view provide users or business partners a good collaboration environment to solve their problem and a platform to share learning.
- TSI have high integration and compatibility, where the time series data that flow into the system can be come from various sources, any sensor or device that play a part in the IoT. REST query API provided by TSI enable user or companies to integrate the data analysed into their apps for other usage.
- Nearly real time data analysis, when the event data from the connected source flow into the system, data can be view though a explorer provide by the TSI and query the data within the TSI. During the real time analysis, alert will be trigger as the anomalies detected by the system.

Drawbacks

- Although TSI have two available tiers S1 and S2 which allow users to push data into the system but there is a limit on it. TSI allow users to push one millions of event data into S1 per Ingress Unit in a day, while S2 allow 10 millions of data flow in. If user a not aware on it, only the first millions of data are available in the system. User are require to Ingress Units otherwise user have to push data into system few day advance to prevent the issue.
- The throughput of system affect by each event size, the size of the event specify in the TSI also become a problem for user. The first tier S1 allow one millions of event push in to the system but each event size will be calculated in 1KB, which mean that if an event push into the system that have 1.1KB will be calculated as 2 events and reduce the throughput. (Vunvulea, 2017).

2.3 IBM (Watson)

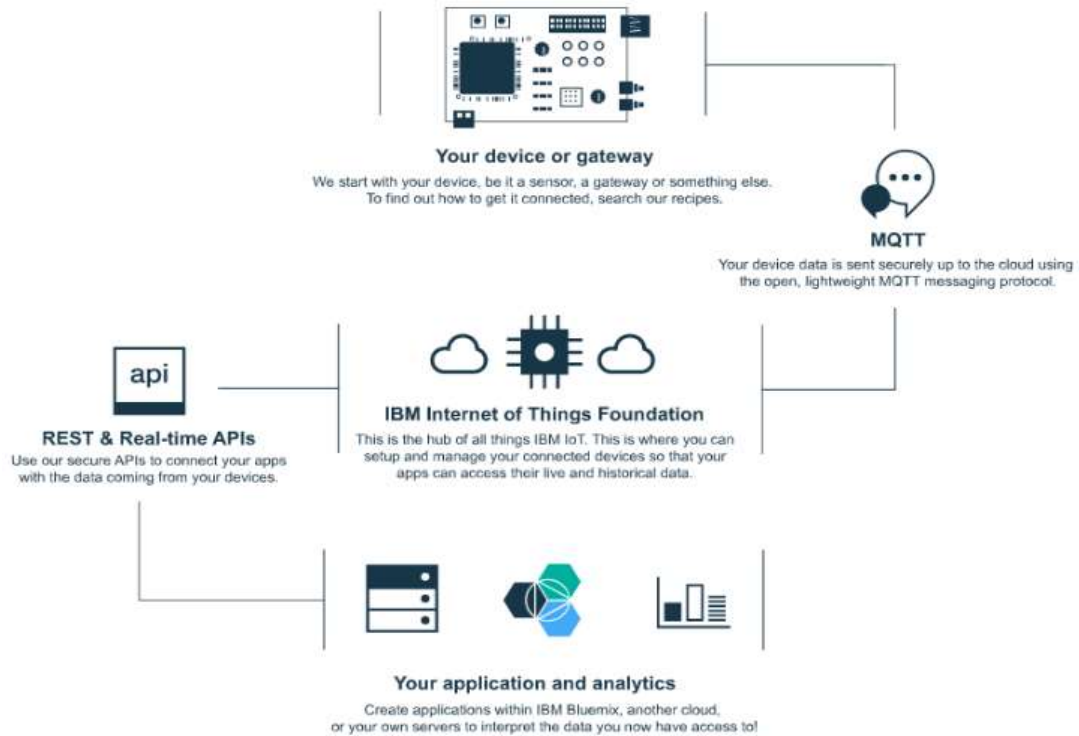


Figure 2- 3 IBM IoT overview. (Pistilli 2016)

IBM Watson IoT is a powerful cloud hosted service, it help to determine the relationships and investigate the correlations among different data set. It enable user to interact with the data and provide the predictive analytics automatically, enable user to spot the tread of data and gaining an insight of data in dashboard. (Get business insight in minute n.d.). To gain a more complete insight of the business, Watson Analytic will compare the result from the analysis with other data sources. Watson IoT platform contain Watson APIs, resource from rich developers to assist enterprise gain IoT insight. Watson IoT manage the unstructured IoT data by identify, combined and transform the data source into a well define asset-based structure, the result will then be presented though a better visualization. (IBM Watson Analytic for Social Media n.d.). Watson allow real-time analytic of the raw data and applying the dashboards to get graphical representation of the data set, user able to get a quick understanding and overview of the information hidden in the data. Watson IoT help companies to govern their security and control the risk to maintain the IoT ecosystem, though anomaly detection or data validation from the analysis.

Strengths of IBM Watson

- Ability and the strength to process the unstructured IoT data. Unstructured data can that collected from the user or companies can simply upload into the cloud, the hidden information and trend of data will be detected automatically by their solution on the cloud, suggest with various extra analysis.
- Strong cognition where user able to gain insight and understand the hidden information from different data source, e.g. data from video, image or even unstructured textual data.
- Secure and maintain IoT ecosystem, help to control and manage risk using dashboards to detect anomaly and alert to monitor various connections.
(Watson IoT Platform n.d.)
- Link data from different source into the decision platform to provide business support and assist business grow.

Drawbacks

- If user need to get all the useful feature and more storage and data source, user require to pay 30 US dollar every month. (IBM Watson Analytic n.d.). IBM Watson require a very high switching cost, and it target toward those larger company or organization that able to afford Watson.
- Maintenance also a problem of Watson and it need users to spend more hours and period to integrate Watson and their services into the user's company.
- IBM Watson require users to put more effort, spend much more time to learn their system, its feature and function in order to use it in its full potential.
- Data flow into the system does not process directly, it require to identify, combine and transform the structure again first before process the data analysis. (IBM Watson n.d.).

Engines / Tools On-premise Approach

2.4 InfluxDB

InfluxDB is an open-source application that written in Go programming language, it is a high performance time series database that used mainly to store the huge number of time based data, log data and data that collected from the IoT sensor. Influx also concern on the query result on the dataset, InfluxDB has provide a time series platform that design to manage events data and metrics in real time. It is a database that store platform in InfluxData and as TICK stack, which represent for Telegraf, InfluxDB, Chronograf and Kapacitor. (Aasen, 2017). It provide a SQL like query language known as InfluxQL, easier for those SQL users to use InfluxDB because those known syntax or syntax that people frequently used able to run in InfluxDB to collect data.

2.5 Graphite

Graphite is an open-source and real-time monitoring time series metrics platform that written in python programming language. The built-in web interfaces are supported by Graphite enable user to browse their timely data metrics. It similar to the database but does not collect any metrics by itself and it relies on the Carbon as a collector to collect the metrics. When the metrics received by the Graphite, transform, combine and query of metrics can be done in real time. Graphite consist of three parts, which is Carbon act as a collector and listen for the timely data, a database known as Whisper and a web app then help to provide graphics upon request. (Taylor, 2017)

2.6 OpenTSDB

OpenTSDB is a scalable time series database written in Java programming language, where it able to store a huge volume of time series data well, it is built on top of HBase and Hadoop. OpenTSDB can be used as a tool to monitor various system, it support and enhance the process of storing and analyzing the raw data that captured from the IoT sensor or other sources. Timely data that flow into the database will be optimize by the combine the data with similar time series to reduce the storage space needed. It allow users to interact with the OpenTSDB by running more time

series database, allow users to interact with time series database through HTTP API. (OpenTSDB n.d.)

2.7 Elasticsearch

Elasticsearch is a highly scalable search engine. It allows users to store, search and retrieve a huge volume of data in nearly real time. It is free and allows users to run Elasticsearch on-premise. It is useful when working with big data because it allows users to query in a very high speed, it can run hundreds of servers with petabytes of data. Elasticsearch is highly tolerant to failure and very flexible on the data type used. (Melo, 2018) Elasticsearch is one of the NoSQL distributed and document-oriented databases, it provides a lot of APIs for users to store, search and manage documents in various structures. It stores data in the JSON document format and provides an index to the document data. (Makadia, 2017)

2.8 Grafana

Grafana is an open-source software that has feature-rich dashboards used to display various metrics on the web browser. It became one of the popular visualization tools to monitor large amounts of event data. Grafana is nice because it is easy to set up, maintain, easier to use and display metrics in a good style. It supports Graphite, InfluxDB, OpenTSDB, more than 30 open-source and types of data sources. (Kili, 2018).

2.9 Kibana

Kibana is an open-source visualization platform and an exploration tool for time series data analytics and Elasticsearch. It is popular with its search engine and analytics, users are able to view and interact with the data that is stored in Elasticsearch. Kibana helps users to understand the large volume of data through visualizing data in various charts, graphs or tables. It allows users to create and share dashboards that display Elasticsearch in real time. (Kibana n.d.)

2.10 Limitation and weakness of the cloud service method

Base on approaches from Amazon, IBM and Microsoft Azure, the IoT analytic solution are perform under their system and services provided on the cloud. However they have some common characteristic and problems.

- Their approaches and method are enterprise and not an open source platform. Their system will evolve slower as compare to the open source platform, the system's solution are transparent any may come with unforeseen limitations.
- Lack of flexibility in term of the security authentication method and the data storage. Users cannot choose or change the authentication method or any security function to protect and secure their event data as they only allow to use the security system provided. Users only allow to use the database or data store that is supported by the system to store their event data.
- The whole system is huge and too complex, it take time and effort to learn all the function in order to use the system in full potential. High complexity of system difficult to perform any custom function.
- Users have to subscribe the services provided in order to get the support from their system on cloud which is troublesome and require user to pay on demand.
- The IoT analytics are perform and done on their private cloud only, user cannot analyse the IoT data on premise and perform the custom function, less control over the event data.

As a result, although the cloud service approaches and the cloud service method have their own strengths in analyzing the IoT data, however they are having the same limitation that cannot perform and handle the custom query function well and it is not suitable to be the IoT analytic platform used to solve the custom query function.

2.11 Comparison between Graphite, InfluxDB, OpenTSDB and Elasticsearch

Name	Graphite	InfluxDB	OpenTSDB	Elasticsearch
Open Source or enterprise	Open Source	Open Source	Open Source	Open Source
Predefined data types	Numeric data	Strings and numeric data	Strings and numeric data	Core, Complex(JSON), Geo and Specialized (IPv4), data type
Support of SQL	none	InfluxQL, SQL-like query language	none	Elasticsearch SQL, SQL-like query language
Access method and API	HTTP API and sockets	HTTP API and JSON	HTTP API and Telnet API	HTTP API and JSON
Method storing different data on different nodes	none	Though sharding	Sharding base on HBase	Though node Data path sharing
Access control on users account	None	Though simple right management	none	Though Role-based Access Control
Supported programming language	(Node.js) in JavaScript Python	Go, Java, R, Haskell, Ruby, JavaScript, PHP, Python, (Node.js) in JavaScript, and others	Go Java R Python Ruby Erlang	Java, Python, Ruby, JavaScript, Apache Groovy, C# and others

Table 2- 1 Comparison between Graphite, InfluxDB, OpenTSDB and Elasticsearch

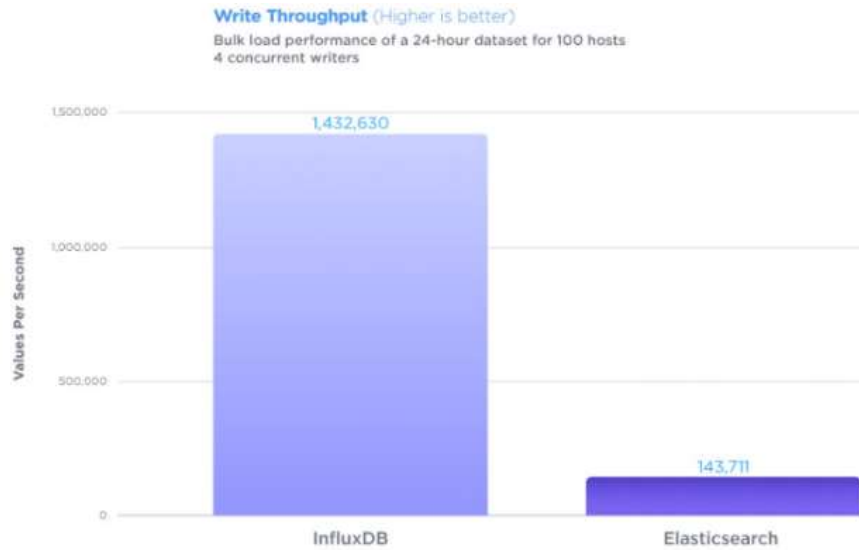


Figure 2- 4 Comparison of write performance for InfluxDB and Elasticsearch

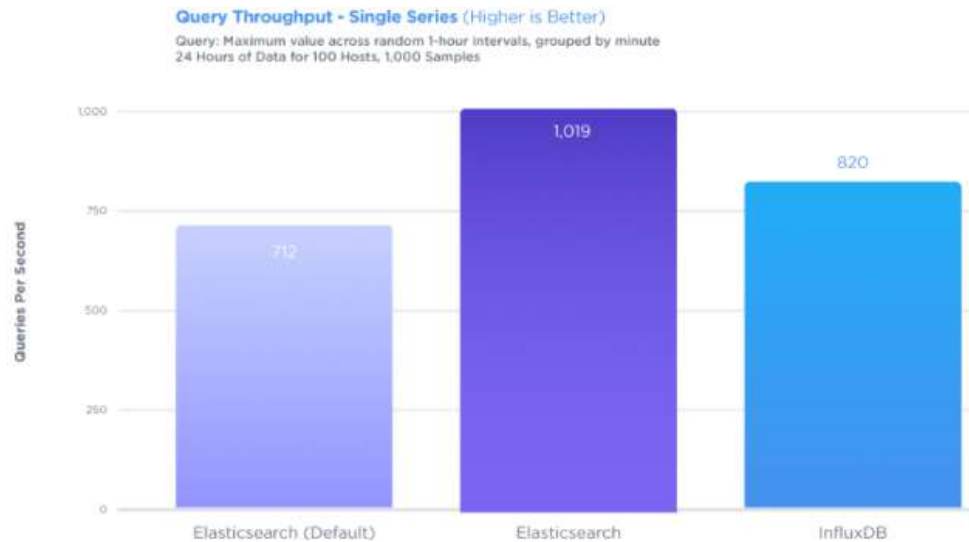


Figure 2- 5 Comparison of the query throughput for InfluxDB and Elasticsearch

From the comparison above, we can see that InfluxDB and Elasticsearch are much stronger from the rest of the database because they able to provide SQL-like environment easier for SQL users to use the tool, provide simple access control on user accounts and supported the most languages. The write performance of InfluxDB is 9 time faster than OpenTSDB and Elasticsearch. However, Elasticsearch having a better query throughput as compare to the InfluxDB. (Churilo, 2018) We can say that

both InfluxDB and Elasticsearch are the strong and powerful database as both of them have their own merits.

2.12 Comparison between Grafana and Kibana

➤ Dashboards Editing and Visualizations

Grafana and Kibana provide many useful feature and enhance visualization capability.

- Both Grafana and Kibana are versatile, they able to filter the data in real time. They support various visualization though graph, table and metric and allow user to do editing on the dashboard. Grafana has a greater and wider option than Kibana in dashboard customization, easier for user to use and make change.

➤ Setup, installation and configuration

Both Kibana and Grafana able to support and install in various platform include Mac, Linux, Window and others.

- Grafana – Easier to be handle because it can be configured by using the ini.file
- Kibana – More difficult to handle as it using its syntax-sensitive YAML configuration file.

➤ Data source support

- Grafana – Work well with different data source including Graphite, Elasticsearch, InfluxDB and others additional source and plugin.
- Kibana – Only work well with the Elasticsearch, not suitable to support other data type and sources.

➤ Querying

- Grafana – Query Editor is query language use by Grafana, it is compatible for different data source and different syntax will be us according to the type of data source.
- Kibana – Lucene is a query language use by Kibana, it is not intuitive and involve certain learning curve. (Yigal, 2018)

Both Kibana and Grafana are useful visualization and monitoring tools but there are use to handle different type of data type or user case. Kibana is more suitable use in analyze the log data, while InfluxDB and Elasticsearch able to work well with Grafana to perform the metrics analysis. Kibana and Grafana also having a common limitation, they cannot perform and handle the custom query function and visualize the data. Grafana have more plugins available today to provide more feature to the editing dashboard and enhance the analytic system.

In this project, Elasticsearch will be selected as the data sources for Grafana to visualize and monitoring the custom function, because this analytic platform can be done on-premises, cost effectively, more flexible in terms of data sources use and security authentication. The limitation and problem of Grafana can be resolve by develop a program that seat between the Grafana and data source, the program redirect the query request and further process by other applications or programs, result return again to Grafana for display. The program to be develop in this project will improve the extension functionality of the analytic platform.

Chapter 3 System Design

This chapter is description the overview of the system,

3.1 Use case design

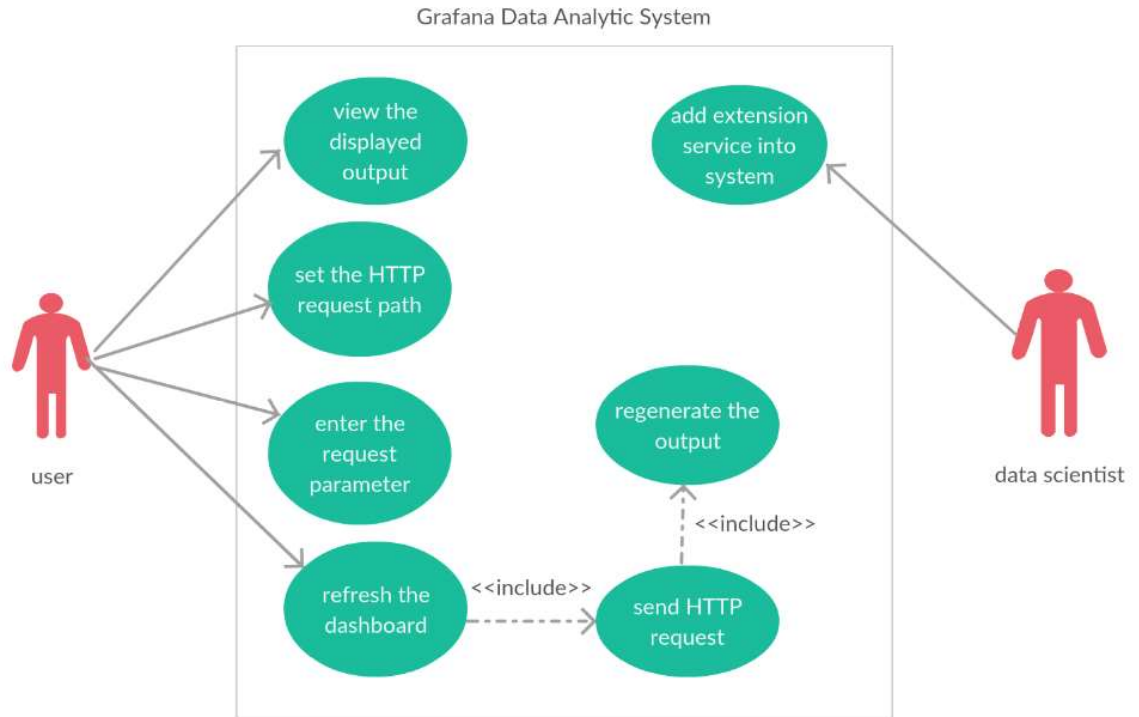


Figure 3- 1 Use case design of Grafana Data Analytic System

The use case diagram above showing how external user interact with the system. User can view the displayed output in the Grafana, enter the request parameter, refresh the dashboard for Grafana to refresh each and every panel. Data scientist can add extension service or advance analysis function into the Grafana analytic platform.

3.2 Use Case Description

Use Case ID	UC001	Version	1.0
Use case	Refresh the dashboard.		
Aims	To allow system to generate a new output on the dashboard.		
Actor	User		
Trigger	User create a dashboard in the Grafana User Interface, set the panel title of the dashboard in the view mode.		
Precondition	User must enter the correct HTTP request path and correct request parameter, in order to generate the result on the dashboard.		
Main Flow	<ol style="list-style-type: none"> 1 User create a dashboard from the Grafana user interface 2 User must change the panel title of the dashboard into the edit mode 3 User must enter the correct HTTP request path 4 User must set the correct request parameters 5 User click on the refresh dashboard button 6 The system will generate a new output and display on the dashboard again 		
Alternative Flow	5.1 System display error message		
Author	Gan Kian Yong		

Table 3- 1 Refresh the dashboard use case description

Use Case ID	UC002	Version	1.0
Use case	Enter the request parameters		
Aims	Allow users to enter and use different kind of parameter to generate various output.		
Actor	User		
Trigger	User change the panel title of the dashboard into the edit mode.		
Precondition	User have created a dashboard panel from the Grafana User Interface		
Main Flow	<ol style="list-style-type: none"> 1 User create the dashboard from the Grafana User interface 2 User must change the panel title of the dashboard into edit mode 3 User allow to set and enter the parameters via the JavaScript column 		
Alternative Flow	3.1 System display error message		
Author	Gan Kian Yong		

Table 3- 2 Enter request parameter use case description

Use Case ID	UC003	Version	1.0
Use case	Set the HTTP request path.		
Aims	To allow user enter the HTTP request path.		
Actor	User		
Trigger	User change the dashboard panel title into the edit mode.		
Precondition	User have created a dashboard panel from the Grafana User Interface		
Main Flow	<ol style="list-style-type: none"> 1 User create the dashboard from the Grafana User interface 2 User must change the panel title of the dashboard into edit mode 3 User can enter and set the HTTP request path into the column 		
Alternative Flow	3.1 System display error message		
Author	Gan Kian Yong		

Table 3- 3 Set HTTP request path use case description

Use Case ID	UC004	Version	1.0
Use case	View the displayed output		
Aims	To allow users understanding the information hide in the data.		
Actor	User		
Trigger	User create a dashboard in the Grafana User Interface, set the panel title of the dashboard in the view mode.		
Precondition	User must enter the correct HTTP request path and correct request parameter, in order to generate the result on the dashboard.		
Main Flow	<ol style="list-style-type: none"> 1 User create a dashboard from the Grafana user interface 2 User must change the panel title of the dashboard into the edit mode 3 User must enter the correct HTTP request path 4 User must set the correct request parameters 5 User click on the refresh dashboard button 6 The system will generate a new output and display on the dashboard again 7 User change the dashboard into the view mode 		
Alternative Flow	5.1 System display error message		
Author	Gan Kian Yong		

Table 3- 4 View the display output use case description

Use Case ID	UC005	Version	1.0
Use case	Add extension service into system		
Aims	To allow data scientist add more application to enhance the data analytic platform.		
Actor	Data Scientist		
Trigger	Open and access to the pythonFile folder.		
Precondition	Data Scientist must add their application and files into the correct path		
Main Flow	<ol style="list-style-type: none"> 1 Data Scientist must access to the HTTP request path 2 Data Scientist open the pythonFile folder 3 Data Scientist add their extension service file into the folder 		
Alternative Flow			
Author	Gan Kian Yong		

Table 3- 5 add extension service to system use case description

3.3 Activity Diagram

Grafana Analytic Module

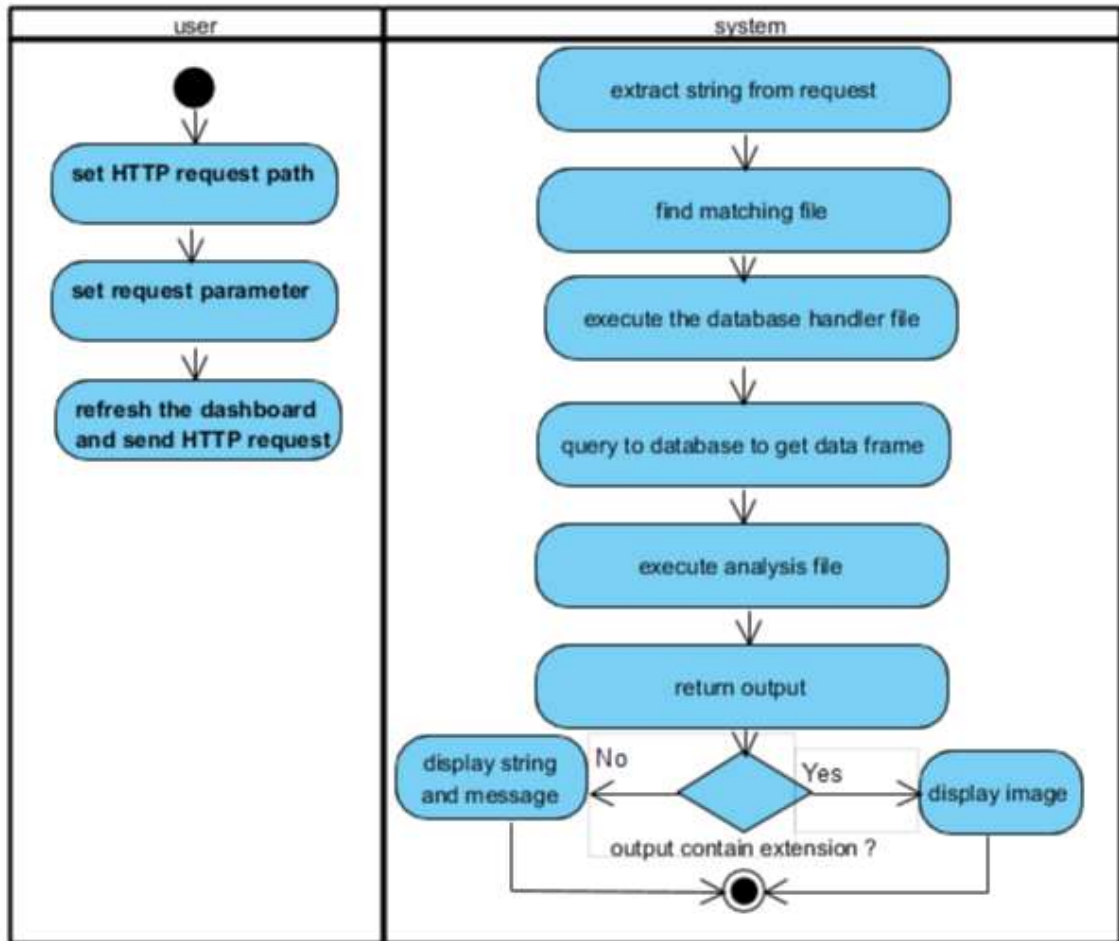


Figure 3- 2 Activity Diagram Grafana analytic module

3.4 Sequence Diagram

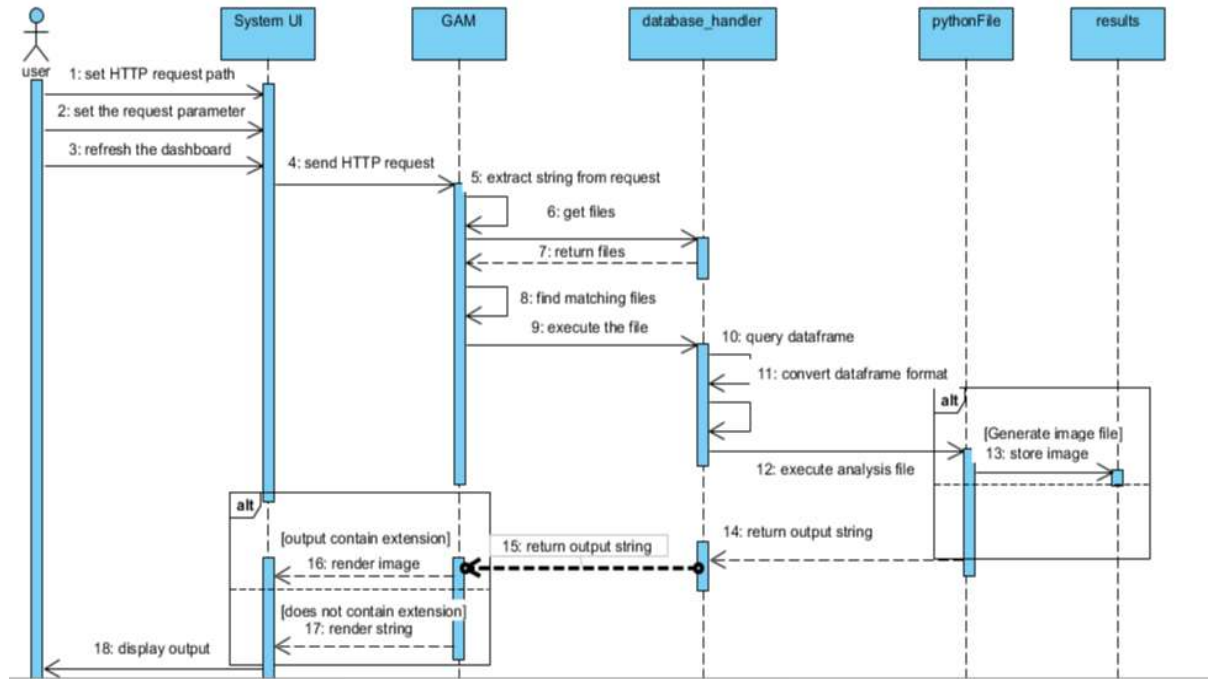


Figure 3- 3 Sequence Diagram of Grafana Analytic Module

3.5 Communication Diagram

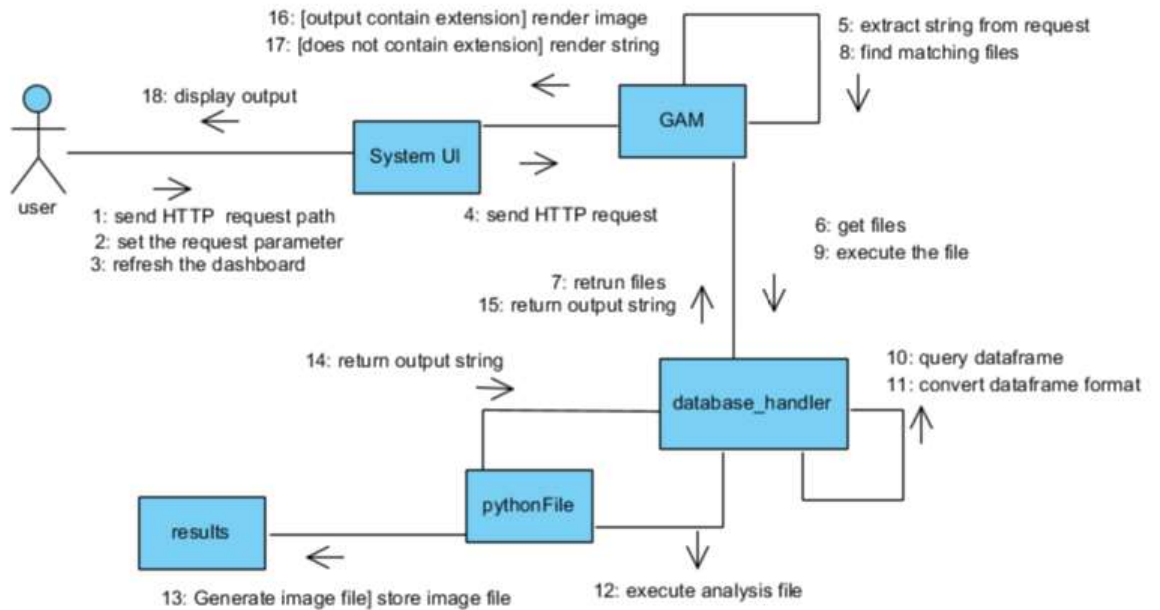


Figure 3- 4 Communication diagram of Grafana analytic module

3.6 Timeline

Gantt Chart

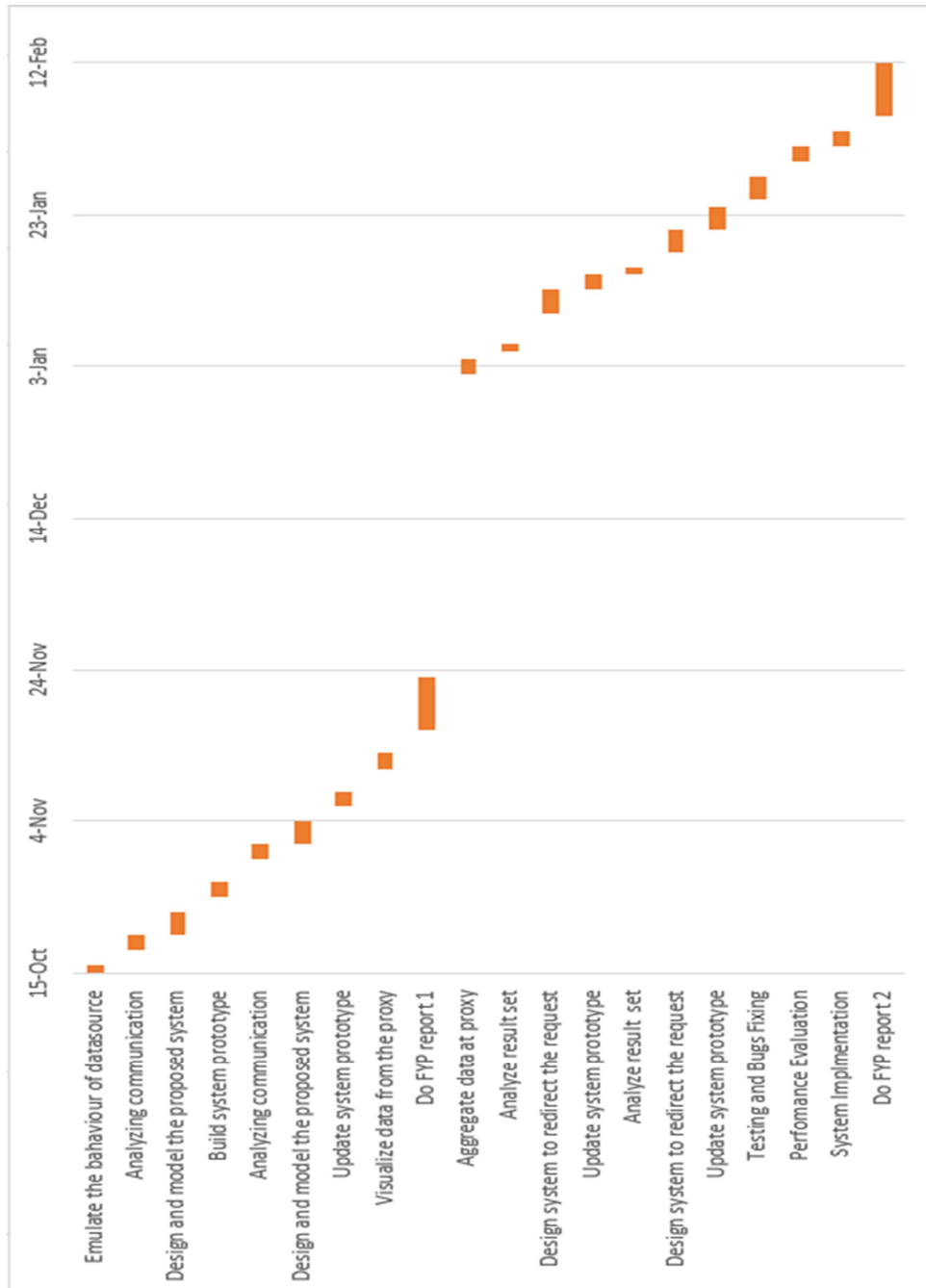


Figure 3- 5 Gantt Chart of FYP1 and FYP2 Gantt

Chapter 4 System Implementation

4.1 Methodologies

Rapid Application Development, prototyping development is chosen as the development methodology for this project and emphasis on rapid creation of prototyping development, shorten the product delivery time. (Ghahrai, 2017).

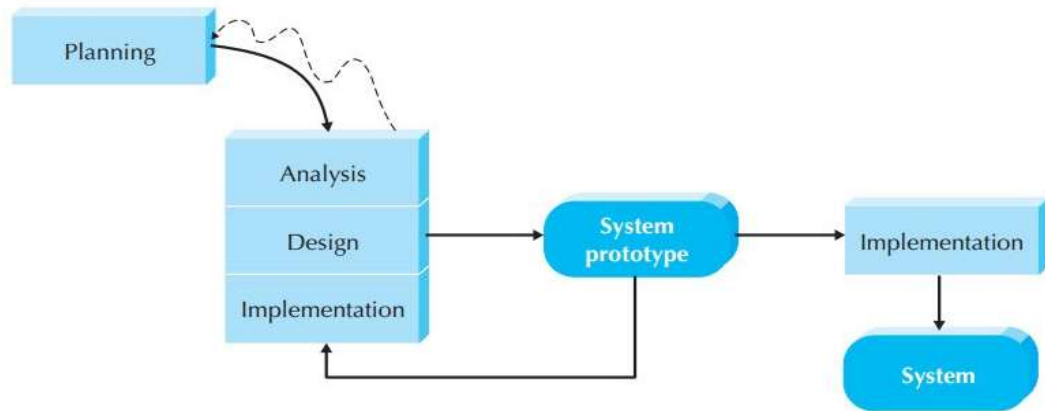


Figure 4- 1 A prototyping-based methodology

(Intro System Analysis and Design n.d.)

The analysis, design and implementation stage will be go through repeatedly to improve the prototype until the system prototype fulfil all the necessary function. Three prototype will be develop throughout the project. The first prototype able to get the HTTP request from the Grafana, analyse the string query append on the request. Execute the suitable python file to make a query and the data frame from the data source. The next prototype able to send or redirect the data frame which query from the data source to another program to conduct analysis. The result will send back to Grafana for display the final output. The third prototype further improve the analytic platform in other kind of advance analysis. The output of the analysis can be a string or an image, make the analytic platform stronger, the output of the analysis not limited to the number or string only.

4.2 Tools to use

-Development Tools

- Linux Ubuntu - An open source operating system to construct and develop the proposed system.
- Visual Paradigm - a software that used to construct and draw the UML diagrams to model the system and their flows.

-Visualization tools

- Grafana plugin (AJAX) – a plugin that use to generate and send the HTTP request to data source or a server, display the result on the editable dashboard .

-Data source

- Elasticsearch - A database that build on top of Apache Lucene. It can search search and store index document files in various structure in real time, act as another data source for Grafana

-Python Libraries

- Pandasticsearch – provide API connect to Elasticsearch server, and query to get get the dataframe.
- Influxdb – provide API connect to the InfluxDB server and query to get dataframe from the database.

-Remote computing tools

- MobaXterm – a software application that help to handle the remote computing job in a simple fashim and manner. It allow user use SSH to connect to the remote server.

4.3 Processing data for visualization

Grafana allow users to create different kind of panel from various data source, a powerful analytic platform to allow user to visualize and monitor the data.

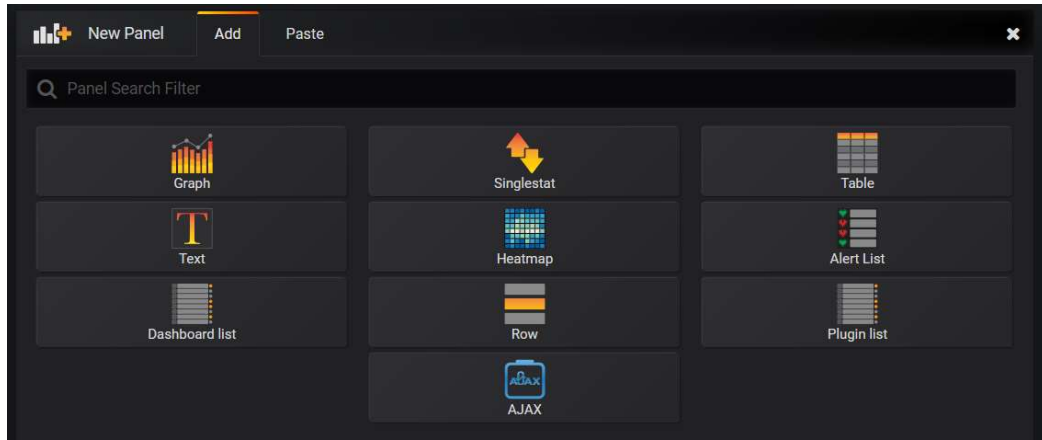


Figure 4- 2 Different panels available on the Grafana

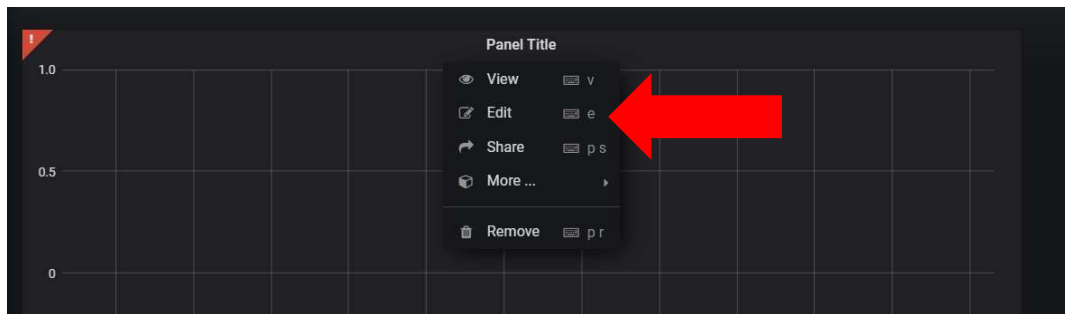


Figure 4- 3 Available option for each panel

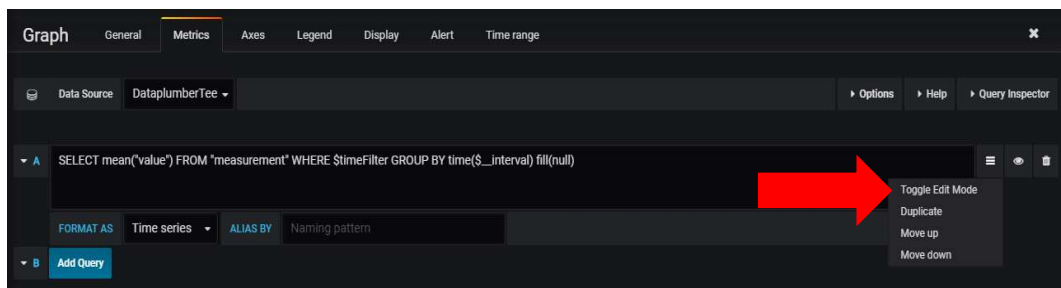


Figure 4- 4 Panel in the edit mode

From the figure above, “toggle edit Mode” selected, to view and make change on the query request. Make change on the primitive query request to redict the request to conduct advance analysis on premise. A module DataPlumbersTee.jar will be executed to redirect th query from Grafana.



Figure 4- 5 Modified query request in the panel

Append the file name of the data source and the name of the advance analysis in front of the query request in order to perform advance analysis or custom query function.

```
def process_incomingQuery(self, incoming_query):
    pre_list = unquote(incoming_query.strip()).replace("+", " ").split(";") ## only first query require
    query_list = pre_list[0].split()
    list_size=len(query_list)

    analysis=query_list[0]
    fromTime=-1
    toTime=-1

    if (list_size==13):
        fromTime=query_list[8][:-2]# remove last two character "xxx ms"
        toTime=datetime.datetime.now().strftime('%s')+ "000"

    elif(list_size==15):
        timeUnit= query_list[10][ len(query_list[10])-1 ]
        number= int((query_list[10])[:-1] )
        toTime=(datetime.datetime.now())

        if (timeUnit=='d'):##days
            fromTime = (toTime - datetime.timedelta(days=number)).strftime('%s')+ "000"

        elif (timeUnit=='h'):##hours
            fromTime = (toTime - datetime.timedelta(seconds=number*3600)).strftime('%s')+ "000"

        elif (timeUnit=='m'):##minutes
            fromTime = (toTime - datetime.timedelta(seconds=number*60)).strftime('%s')+ "000"

        toTime=(datetime.datetime.now()).strftime('%s')+ "000" ## convert all to UnixTime str first

    elif(list_size==17):
        fromTime=query_list[8][:-2]# remove last to character "xxx ms "
        toTime=query_list[12][:-2]

    return analysis,int(fromTime),int(toTime)
```

Code Segment 4- 1 process the incoming query request

This segment of code process the in coming query request, split the query request into segments, based on the number of segment to determine the times and indentified the analysis file require to be executed.

```
def get_data(self, analysis, startTime, stopTime):
    sys.path.insert(0, '/var/www/html/pythonFile/advanceQuery/')
    f= __import__(analysis) # import the analysis.py file

    dataHeader,dataList=f.function(int(startTime),int(stopTime))
    return dataHeader,dataList
```

Code Segment 4- 2 get data from the analysis function

After the program identified the times and the analysis file require to execute. Another function `get_data` was invoke to query data from the data souce and perform the necessary analysis. The output of the analysis function will be return and store into two list strcuture.

```
def display_multidataframe_to_grafana(self, title, out_dataframe, row_column = 0, table = N
    json_statement_array = []

    for rc in range(0,row_column):
        json_values_list = []

        for time, dataCol in out_dataframe.iterrows():
            json_value_point = []
            json_value_point.append( pd.Timestamp(time , tz = 'Asia/Kuala_Lumpur').value)
            json_value_point.append(dataCol[rc+1])
            json_values_list.append(json_value_point)

        json_columns_array = []
        json_columns_array.append("time")

        json_series = {}
        json_series["name"] = "Advance"
        json_series["columns"]=["Time",title[rc]] #1 col for 1 data
        json_series["values"] = json_values_list
        json_series_array = []
        json_series_array.append(json_series)

        json_statement = {}
        json_statement["series"] = json_series_array
        json_statement["statement_id"] = rc
        json_statement_array.append(json_statement)

    json_result = {}
    json_result["results"] = json_statement_array

    print (json.dumps(json_result).replace("'",'').replace("NaN","null"))
```

Code Segment 4- 3 show the dataframe to the Grafana

When the result was return from the analysis function, the output data that store in the list will be convert into the dataframe format. The function `display_multiframe_to_grafana` will be invoke, the dataframe will be arrange accordingly and convert into the well formatted JSON format and display to Grafana

```
{"results": [{"series": [{"values": [[1522771200000, 0.6569257304534811], [1522857600000, 0.6569257304534808], [1522944000000, 0.6566455801478162], [1523030400000, 0.6566455801478169], [1523116800000, 0.6566455801478173], [1523203200000, 0.6567069689622612], [1523289600000, 0.6567490700159164], [1523376000000, 0.6567490700159164],
```

Figure 4- 6 show the data display to Grafana in the JSON Format.



Figure 4- 7 Data visualization on the panel from the JSON string

Grafana display the graph based on the given JSON string, grafana allow user to refresh the panel to generate the new output again.

4.4 Visualize data from advance query function

AJAX is one of the plugin available for Grafana, this panel able to send a HTTP request to the server and render an image on the panel. This panel extend the functionality of this analytic platform, the output of the analysis are not limited to the number and string only, it can show a diagram, figure or an image.

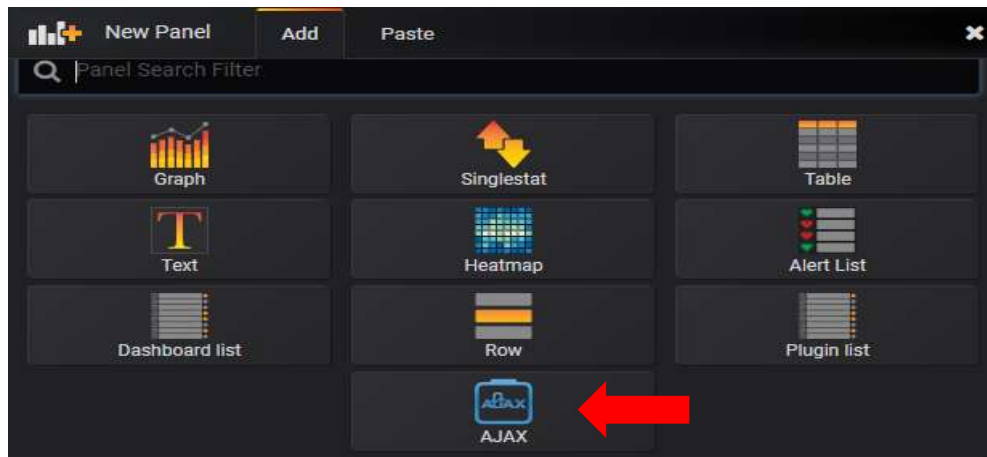


Figure 4- 8 AJAX panel on grafana

AJAX panel is a useful plugin for Grafana, we make use of this plugin to render image, figure or diagram on the panel to display the output of the advance analysis function.

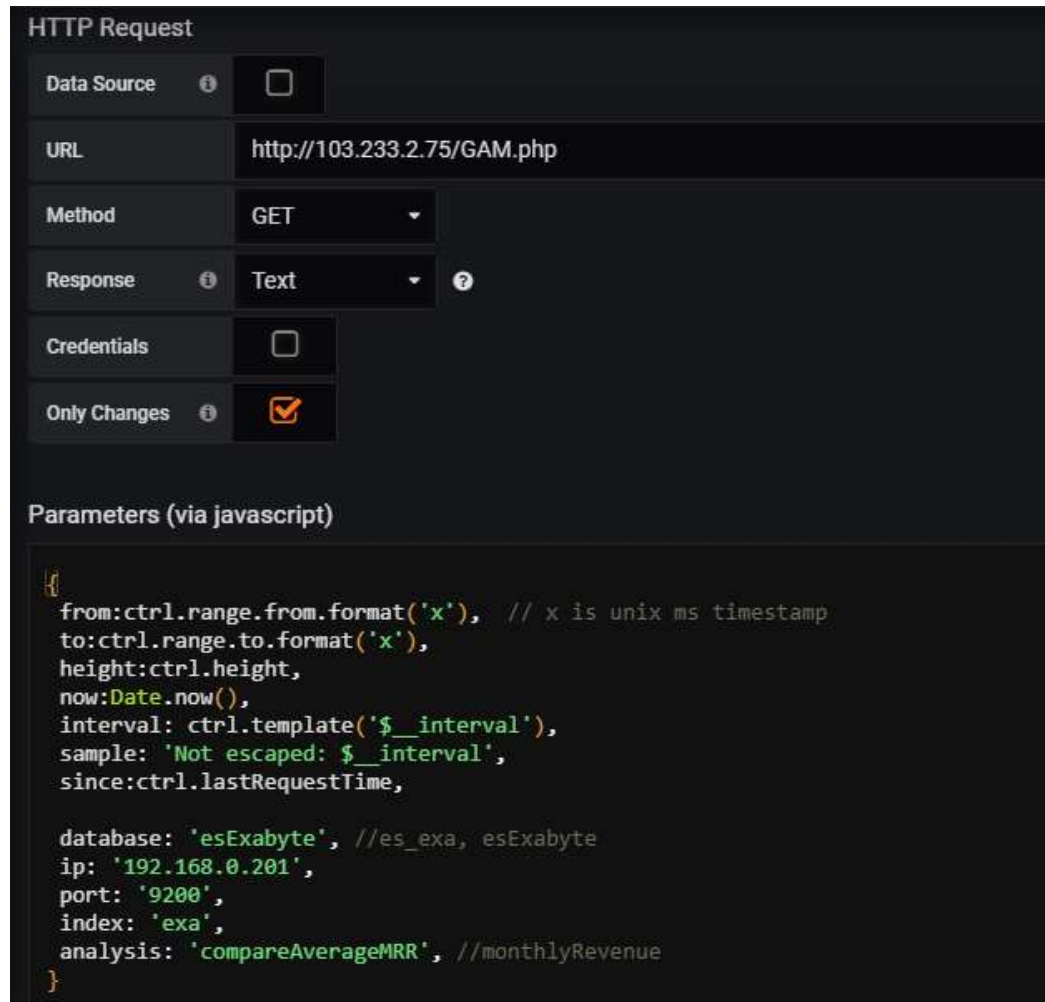


Figure 4- 9 show the environment in the edit mode of the AJAX panel

This panel allow user to send a HTTP request, direct to the URL and add the parameters that append on the HTTP request via the javascript.

```

http://103.233.2.75/GAM.php?from=1396450402647&to=1554216802648&height=233&now=1554216802649&interval=1d&sample
sample=Not%20escaped%3A%20%24__interval&since=1554210194543&database=esExabyte

```

Figure 4- 10 HTTP request sent from Grafana (AJAX)

Figure 4- 9 show the HTTP request send from the Grafana to the GAM php script in the server, and all the parametes append in the request. The parameters are abstract from this HTTP request, used as the argument to conduct other analysis function.

```
$server= $_SERVER['REQUEST_URI'];
$from = $_GET['from'];
$height= $_GET['height'];
$interval= $_GET['interval'];
$now= $_GET['now'];
$sample= $_GET['sample'];
$since= $_GET['since'];
$to= $_GET['to'];

$database=$_GET['database'];
$ip=$_GET['ip'];
$port=$_GET['port'];
$index=$_GET['index'];
$analysis=$_GET['analysis'];
```

Code Segment 4- 4 store the parameters from the HTTP request

The code segment x above show the section of code in the GAM.php script, it used to store each and every important parameter that append in the HTTP request.

```
$files = glob('/var/www/html/database_handler/*.py', GLOB_BRACE);
$database_file=($database. ".py"); //find the matching (since all write in python - add .py on it)

foreach($files as $file) {
    if( $database_file == basename($file)){ //file is the dir of .py

        $output = shell_exec("python $file '$from' '$to' '$database' '$ip' '$port' '$index' '$analysis'");
        $extension=pathinfo($output,PATHINFO_EXTENSION);
        $extension = trim(preg_replace('/\s+/', ' ', $extension)); //remove the whitespace in front and end of the string
        echo '<br>';

        if ($extension=="png"){
            $result_image="http://103.233.2.75/resultImage/$output";
            echo "<img src='$result_image'>";
        }
        else{
            echo $output;
        }
        echo '<br>';
        break;
    }
}
```

Code Segment 4- 5 Determine the analysis file require to be execute and display the images

The section of code above append a “.py “ string to the database parameter and store into a new variable, it used for program to execute the python file later. The php script loop and execute correspond python file, and pass the necessary augments to the python file. The python file (database_handler) will execute the python analysis function and return the output again to this php script. All the python analysis function make use of the

Chapter 4 System Implementation

elasticsearch library to query the data from data source and matplotlib library used to further process the data to generate the result. The python file return the output (image file name) as a string, the php script verify the content of the string, display the image if the returned string contain the “.png” file extension.



Figure 4- 11 result of the AJAX panel

Figure above show the result using the AJAX panel, the image generated from the python analysis files and render on the panel in a short period.

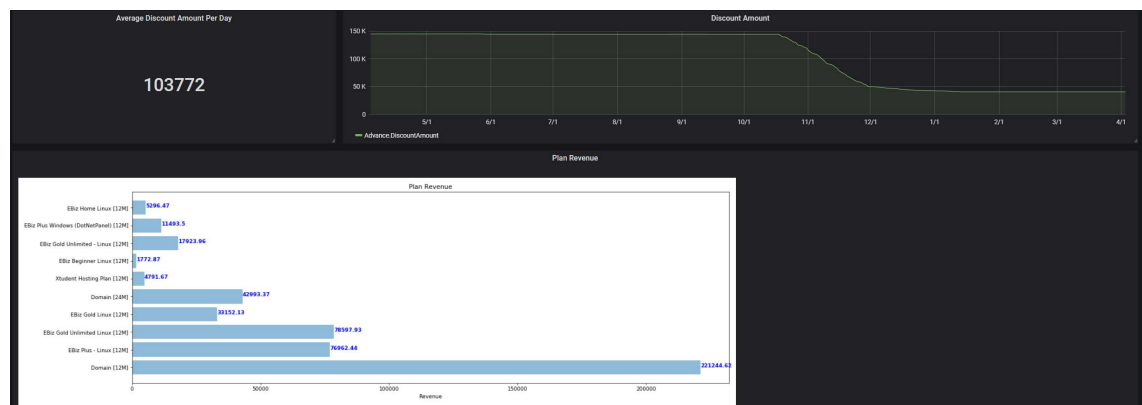


Figure 4- 12 different panels display on the Grafana

Figure above showing that the functionality of the analytic platform have been improved, Grafana able to monitor and visualize various kind of data, conduct and perform different kind of advance analysis function.

4.5 Implementation issue and Challenge

-Difficulty in design the program query to dataframe from different data source

User may not understand how the query work behind the program and don't have any knowledge about the query syntax, the program to be develop must able to generate different query syntax for different data source because different database using different query statement. Design a program and function to handle different kind of query to various database is challenge. Research and review the format of query on different database, find the pattern and format will be easier to construct the design.

-The return type of the analysis file can be vary

As the extension functionality of the analytic platform improve, the data go through more advance analysis. Some analysis like calculating the standard deviation of the data will return a value a output can be display easily in Grafana. However, some other analysis like linear classification, the output is not just a single value only and difficult to display and visualize in Grafana. The problem can be resolve by verifying the output string, if the output string contain any extension it mean that a file have been return, otherwise the output will be a string contain the messages.

-Difficulty in generate and write query statement due to the database store document or data in a complex structure.

Elasticsearch store and support various data type in the database including Core data type, Complex (JSON) data type, Geo data type and Specialized (IPv4) data type. Elasticsearch also able to data and document in a nested and complex structure. It is challenging when constructing the query statement to get the data frame from Elasticsearch.

Review more document and data that store in the Elasticsearch database, try to understand the structure and pattern that store will be easier to design the query.

Chapter 5 Experimental Result

The subsequent experiments are carried out on the server and the Google Chrome is used as the browser to access Grafana.

5.1 Analysis Function and data source support by Grafana

In this experiment, we compare the analysis function of our propose Grafana architecture against the primitive Grafana. For a fair comparison, we using the same kind of data source as the input.

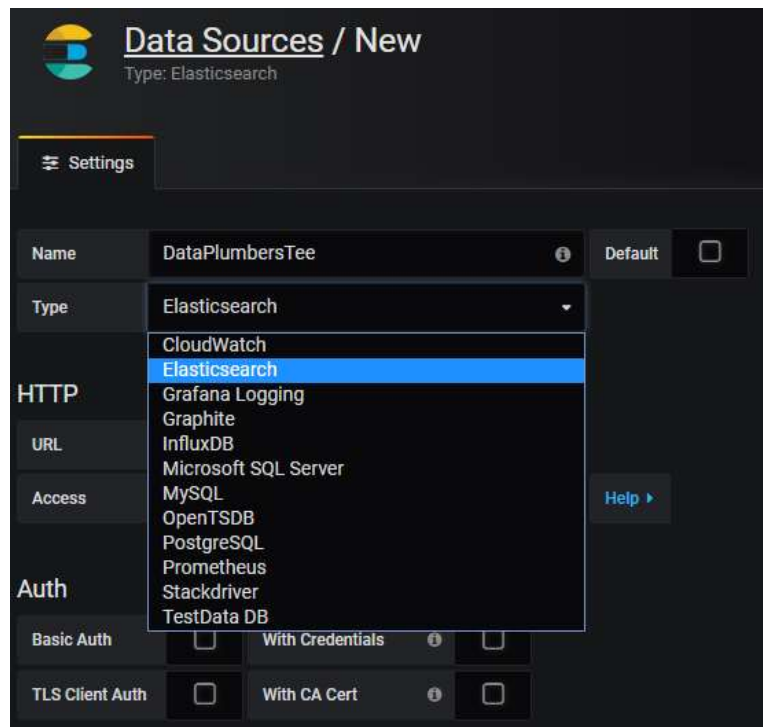


Figure 5- 1 Data sources supported by primitive Grafana analytic platform

From the figure above, the drop down list show the 12 data sources support by the primitive Grafana analytic platform. This figure indicate that the panel created in the dashboard can only query data from these data sources.

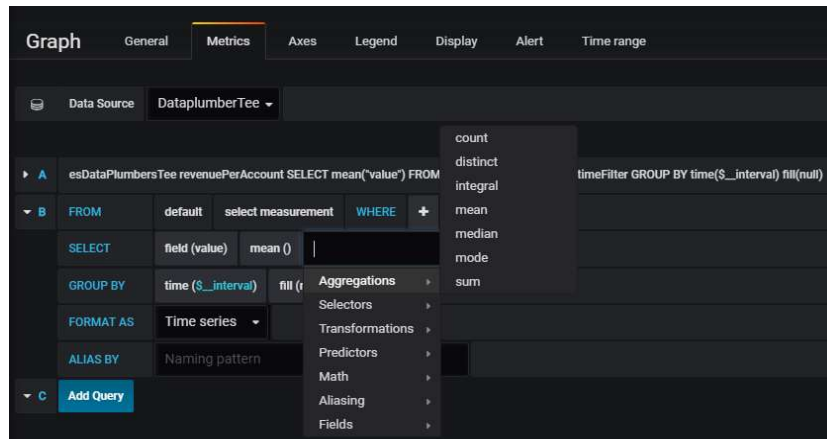


Figure 5- 2 Analysis function available in the panel from primitive Grafana analytic platform

The figure above shows that the function available in the panel, all the function provided in the primitive Grafana analytic platform are the basic analysis function, eg sum, mode, mean , median etc. However, the function available are not enough to conduct those advance analysis and to handle those custom query.

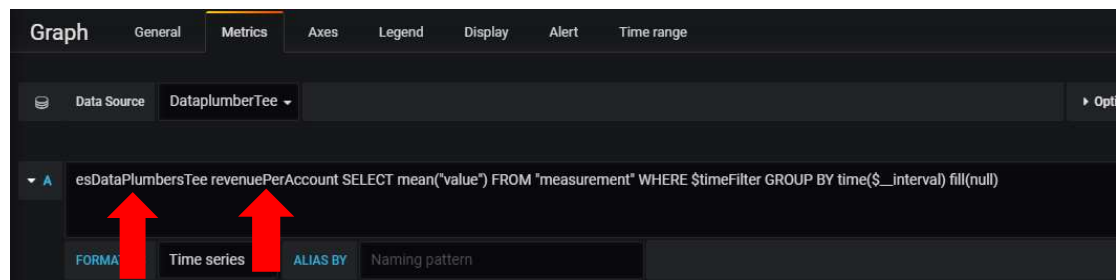


Figure 5- 3 Query used in the improved Grafana analytic platform

Based on the figure above, the primitive query have been modified and 2 new parameter have been added infront of the query. The first parameter will be the name of python files that query data from different data source. The second parameter will be the name of the analysis file to be executed. The improved Grafana analytic platform are more flexible, it can query data more than 12 data sources and conduct more analysis function.

Name	Size (KB)	Last modified	Owner
..			
DataPlumbersTee.jar	15	2019-03-05 ...	root
DataPlumbersTee.log	1	2019-03-15 ...	root
DataPlumbersTee.py	6	2019-03-17 ...	root
DataPlumbersTeeDebug.jar	15	2019-03-05 ...	root
es_exa.py	1	2019-02-25 ...	root
esDataPlumbersTee.py	7	2019-03-22 ...	root
esExabyte.py	1	2019-03-28 ...	root
ESGan.py	1	2019-02-17 ...	root
influxdb.py	1	2018-11-16 ...	root

Figure 5- 4 The name of different data source to handle the query request

Name	Size (KB)	Last modified	Owner
..			
averageCustomerAcquisitionC...	1	2019-03-25 ...	root
averageCustomerAcquisitionC...	1	2019-03-25 ...	root
averageDiscountAmount.py	1	2019-03-25 ...	root
averageDiscountAmount.pyc	1	2019-03-25 ...	root
averageMRR.py	1	2019-03-31 ...	root
averageMRR.pyc	1	2019-03-31 ...	apache
averageRecurringRevenue.py	1	2019-03-25 ...	root
averageRecurringRevenue.pyc	1	2019-03-25 ...	root
averageRevenuePerAccount.py	1	2019-03-22 ...	root
averageRevenuePerAccount...	1	2019-03-22 ...	root
churnRate.py	1	2019-03-28 ...	root
churnRate.pyc	1	2019-03-28 ...	root

Figure 5- 5 The name of the analysis file in the server

Based on the figures above, the primitive Grafana analytic platform able to perform the limited number basic and simple function only. However, the new module developed, improve the Grafana analytic platform and it able to support Grafana to query data from more data sources. In addition, Grafana able to perform more kind of custom query function or advance analysis.

5.2 Visualization function of Grafana

In this experiment, we compare the visualization function available in the primitive Grafana analytic platform against the improved Grafana analytic platform.



Figure 5- 6 different kind of panels in the dashboard

Figure above showing the panels created from the primitive Grafana analytic platform. The primitive Grafana analytic platform only able to visualize data on few type of panel. It can represent the data in graph, table, singlestat, heatmap and text form. However, the primitive Grafana analytic platform unable to visualize data in the form of historgram, pie chart, bar chart etc, this is because every graph of visual that show on the panel are query to the data source, result generate by the data source and returned again to Grafana for display only so its function are limited by the data source.

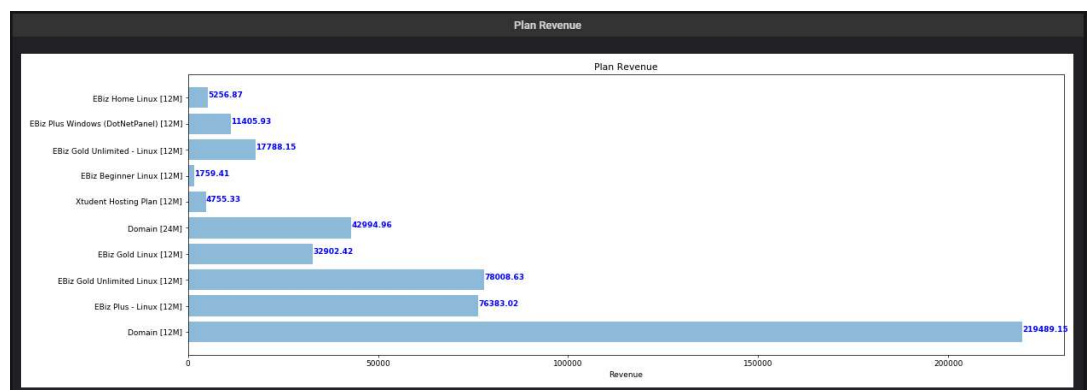


Figure 5- 7 Bar chart in the improved Grafana analytic platform

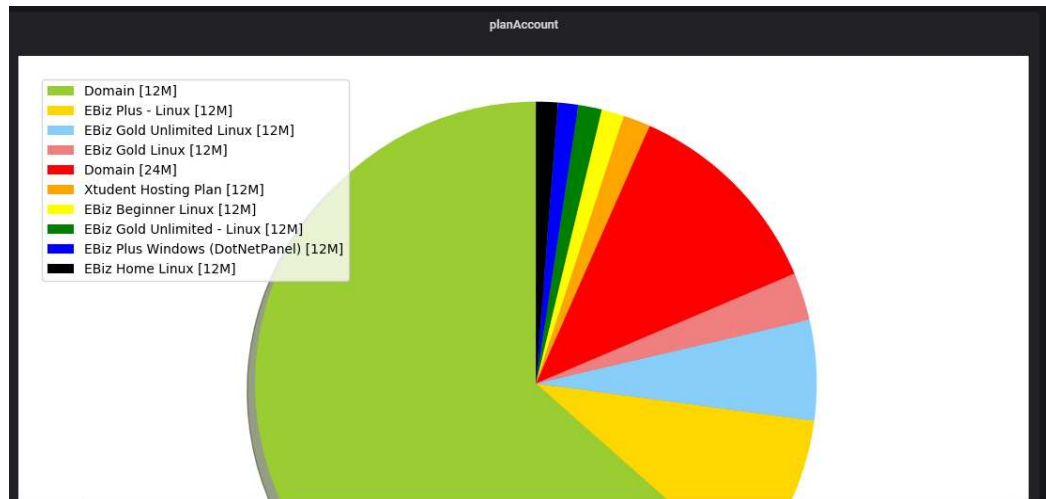


Figure 5- 8 show the pie chart in the improved Grafana analytic platform

From the figure 5- 6, figure 5- 7 and figure 5- 8 above showing that the improved Grafana analytic platform has the ability to visualize more kind of data. The new module developed enhance the Grafana analytic platform, enable Grafana to conduct various kind of custum function and analysis. The analysis function no longer limited by the data source, if the advance query function able to conduct and perform using python script then its result able to display on the Grafana too.

5.3 Extensibility of Grafana

This experiment, we compare the extensibility of both Grafana analytic platform in terms of the compatibility of the new features, functions added to the system.

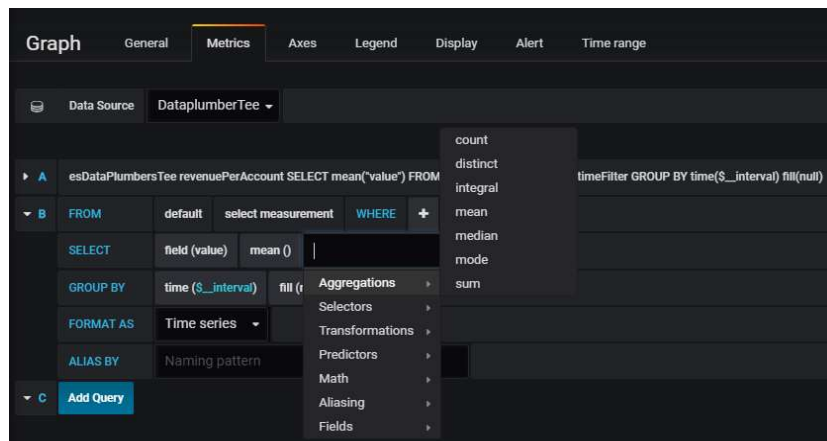
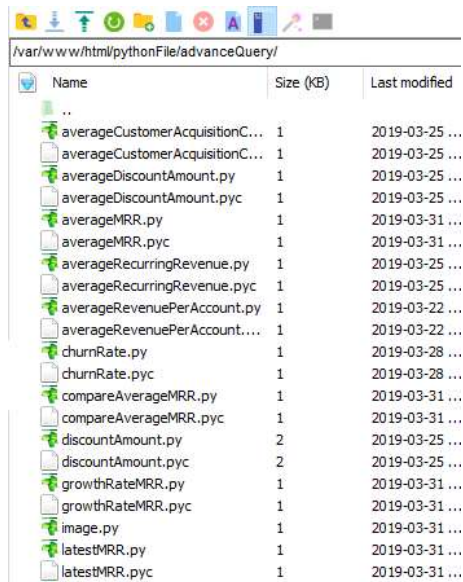


Figure 5- 9 function available in the primitive Grafana analytic platform



The screenshot shows a file manager window with the following table of files:

Name	Size (KB)	Last modified
..		
averageCustomerAcquisitionC...	1	2019-03-25 ...
averageCustomerAcquisitionC...	1	2019-03-25 ...
averageDiscountAmount.py	1	2019-03-25 ...
averageDiscountAmount.pyc	1	2019-03-25 ...
averageMRR.py	1	2019-03-31 ...
averageMRR.pyc	1	2019-03-31 ...
averageRecurringRevenue.py	1	2019-03-25 ...
averageRecurringRevenue.pyc	1	2019-03-25 ...
averageRevenuePerAccount.py	1	2019-03-22 ...
averageRevenuePerAccount....	1	2019-03-22 ...
churnRate.py	1	2019-03-28 ...
churnRate.pyc	1	2019-03-28 ...
compareAverageMRR.py	1	2019-03-31 ...
compareAverageMRR.pyc	1	2019-03-31 ...
discountAmount.py	2	2019-03-25 ...
discountAmount.pyc	2	2019-03-25 ...
growthRateMRR.py	1	2019-03-31 ...
growthRateMRR.pyc	1	2019-03-31 ...
image.py	1	2019-03-31 ...
latestMRR.py	1	2019-03-31 ...
latestMRR.pyc	1	2019-03-31 ...

Figure 5 -10 advance analysis function for the improved Grafana analytic platform

The comparison between primitive and improve Grafana analytic platform is clearly show that the function available in the primitive Grafana analytic platform is limited and difficult to add new function into the system to conduct the analysis. However, the improved Grafana analytic platform has the ability to perform more advance analysis function. Once the advance analysis function developed from the data scientist and add into the server, user can always use it on the Grafana without any complex configuration.

Based on the experiments above, we can conclude that the three main objectives have been achieved. The proposed solution improves the Grafana analytic platform in terms of functionality and the extensibility of the system. Grafana can perform custom query functions and advanced analysis, visualize the custom function on the dashboard. The extension services or functions can be added easily to the Grafana without any complex configuration.

Chapter 6 Conclusion

IoT analytic platform is important for every individuals, because it allow individual to gain more insights, understanding informations hide in the data. In current practise, implementing a IoT solution can be very expensive for in terms of price, complex and take longer time and effort to learn. They are many users facing the problem that the analytic platform cannot solve their custom query function. Therefore the concern of this project is to solve the problems in a way that cost-effective, high flexibility, less complexity and able to use easily for everyone.

Grafana is a well recommended open source and free software tools for monitoring and visualization the user data, it provide a lot of feature and dashboard editing function. However, it has some limitation on handling the custom query and conduct those advance analysis. In an effort to solve the problem and limitation of Grafana, a program that seat at the middle between Grafana and data sources created to support Grafana handle the custom query function. The program have the ability to redirect the custom query function, the custom query problem solve by other applications or programs and return the result back to Grafana for display. The objectives of this project have been achieve where user able to perform the custom query function, visualize the analysis reuslt on the dashboard in Grafana. This project make the Grafana analytic platform independently from the data source, because its functions and features no longer limited by the data source.

Besides, this project improve the extension functionality of the system, because the new analysis function or programs created from the data scientist can be added in to the system though the program. Grafana can visualize and conduct various advance analysis though the program. This project also create a very useful platform for every users, the Grafana analytic module can support user and help them to query data from various database and they are not require to learn and write complex query statement or make any complex configuration.

In future, Grafana analytic platform can support various data sources and visualize different kind of data from advance analysis. Other community, can improve the system by providing more extension services into the system further improve the functionality of the analytic platform, providing access control function to ehance the

Chapter 6 Conclusion

system in term of security perspective, provide the load balancing function to the Grafana, share the workload to prevent the query request overwhelm the server and ensure the system always have the good performance, make system fault tolerance and high availability.

Bibliography

Aasen, G., 2017. *Introduction to InfluxData's InfluxDB and TICK Stack*. [Online]

Available at: <https://www.influxdata.com/blog/introduction-to-influxdatas-influxdb-and-tick-stack/> [Accessed 4 August 2018].

Acreman, S., 2016. *Top 10 Time Series Databases*. [Online]

Available at: <https://blog.outlyer.com/top10-open-source-time-series-databases> [Accessed 6 July 2018].

Admin, W., 2018. *Program of the World Championship WNA nunchaku-do 2018*. [Online]

Available at: <https://www.nunchakubackend.org/e/program-of-the-world-championship-wna-nunchaku-do-2018/> [Accessed 10 August 2018].

Amazon Kinesis Data Analytics, n.d. [Online]

Available at: <https://aws.amazon.com/kinesis/data-streams/> [Accessed 6 July 2018].

Amazon Kinesis Data Streams, n.d. [Online]

Available at: <https://aws.amazon.com/kinesis/data-streams/> [Accessed 6 July 2018].

Amazon QuickSights, 2016. [Online]

Available at: <https://searchaws.techtarget.com/definition/Amazon-QuickSight> [Accessed 6 July 2018].

Bibliography

Azure Time Series Insights, n.d. [Online]

Available at: <https://www.appliedi.net/blog/azure-time-series-insights-a-new-platform-for-enterprise-iot/> [Accessed 8 July 2018].

Azure Time Series Insights, 2018. [Online]

Available at: <https://docs.microsoft.com/en-us/azure/time-series-insights/time-series-insights-overview> [Accessed 8 July 2018].

Churilo, C., 2018. *InfluxDB Markedly Outperforms OpenTSDB in Time Series Data and Metrics*. [Online]

Available at: <https://www.influxdata.com/blog/influxdb-markedly-outperforms-opentsdb-in-time-series-data-metrics-benchmark/> [Accessed 8 August 2018].

Data Source Limits, n.d. [Online]

Available at: <https://docs.aws.amazon.com/quicksight/latest/user/data-source-limits.html> [Accessed 11 July 2018].

Desai, K., 2017. *Build a Visualization and Monitoring Dashboard for IoT Data with Amazon Kinesis Analytics and Amazon QuickSight*. [Online]

Available at: <https://aws.amazon.com/blogs/big-data/build-a-visualization-and-monitoring-dashboard-for-iot-data-with-amazon-kinesis-analytics-and-amazon-quicksight/> [Accessed 11 August 2018].

Flynn, B., 2014. *Amazon Kinesis: Is It the Next Big Real-Time Data Processing Solution*. [Online]

Bibliography

Available at: <http://diamondstream.com/amazon-kinesis-big-real-time-data-processing-solution/> [Accessed 11 July 2018].

Gerber, A., 2017. *Making sense of IoT data*. [Online]

Available at: <https://www.ibm.com/developerworks/library/iot-lp301-iot-manage-data/index.html> [Accessed 2 July 2018].

Get business insight in minute, n.d. [Online]

Available at: <https://www.ibm.com/watson-analytics> [Accessed 8 July 2018].

Heath, N., 2016. *AWS sets its sight on the BI market as it makes Amazon QuickSights generally available*. [Online]

Available at: <https://www.zdnet.com/article/aws-sets-its-sights-on-the-bi-market-as-it-makes-amazon-quicksight-generally-available/> [Accessed 6 July 2018].

IBM Watson Analytics for Social Media, n.d. [Online]

Available at: <https://www.ibm.com/my-en/marketplace/social-media-data-analysis>

[Accessed 8 July 2018].

IBM Watson, n.d. [Online]

Available at: <http://ibmwatson237.weebly.com/advantages--disadvantages.html>

[Accessed 12 July 2018].

Intro System Analysis and Design, n.d. [Online]

Available at: <https://sites.google.com/site/ustcomp2com/cover-page/assignment-003/section-1-introduction>

[Accessed 10 August 2018].

BCS (Hons) Computer Science

Faculty of Information and Communication Technology (Kampar Campus), UTAR.

Bibliography

Kibana, n.d. [Online]

Available at: <https://aws.amazon.com/elasticsearch-service/kibana/>

[Accessed 9 August 2018].

Kibana, n.d. [Online]

Available at: <https://www.elastic.co/guide/en/kibana/current/introduction.html>

[Accessed 9 August 2018].

Kili, A., 2018. *Grafana – An Open Source Software for Analytics and Monitoring*.

[Online]

Available at: <https://www.tecmint.com/install-grafana-analytics-in-centos-ubuntu-debian/> [Accessed 4 July 2018].

Lichtigstein, A., 2017. *Prometheus vs Graphite vs Grafana – A Feature Comparison*.

[Online]

Available at: <https://www.loomsystems.com/blog/single-post/2017/06/07/prometheus-vs-grafana-vs-graphite-a-feature-comparison> [Accessed 10 August 2018].

Makadia, M., 2017. *What Is Elasticsearch and How Can It Be Useful*. [Online]

Available at: <https://dzone.com/articles/what-is-elasticsearch-and-how-it-can-be-useful> [Accessed 10 August 2018].

Melo, M.V., n.d. *A Practical Introduction to Elasticsearch with Kibana*. [Online]

Available at: <https://dzone.com/articles/what-is-elasticsearch-and-how-it-can-be-useful> [Accessed 15 November 2018].

Bibliography

OpenTSDB, n.d. [Online]

Available at: <http://opentsdb.net/overview.html> [Accessed 8 August 2018].

Paxton, A., 2016. *InfluxData Logo*. [Online]

Available at: <https://dribbble.com/shots/2446318-InfluxData-Logo>

[Accessed 10 August 2018].

Pistilli, J., 2016. *5 Things to Know about the IBM Internet of Things Foundation*.

[Online]

Available at:

https://www.ibm.com/developerworks/community/blogs/5things/entry/5_Things_to_Know_about_IBM_Internet_of_Things_Foundation?maxresults=15&lang=ja

[Accessed 8 July 2018].

Pittenturf, C., n.d. *What is Data Visualization and Why is it Important*. [Online]

Available at: <https://data-visualization.cioreview.com/cxinsight/what-is-data-visualization-and-why-is-it-important-nid-11806-cid-163.html>

[Accessed 2 July 2018].

Rijmenam, M.V., 2017. *Why Prescriptive Analytics Is the Future of Big Data*.

[Online]

Available at: <https://www.linkedin.com/pulse/why-prescriptive-analytics-future-big-data-mark-van-rijmenam> [Accessed 3 July 2018].

Bibliography

Rouse, M., 2017. *Data visualization*. [Online]

Available at: <https://searchbusinessanalytics.techtarget.com/definition/data-visualization> [Accessed 2 July 2018].

Shankarnarayan, C., 2018. *Propel your IoT platform to the cloud with Azure Time Series Insights*. [Online]

Available at: <https://azure.microsoft.com/en-us/blog/propel-your-iot-platform-to-the-cloud-with-azure-time-series-insights/> [Accessed 10 August 2018].

Sonja, 2017. *Introduction to Prometheus Monitoring*. [Online]

Available at: <https://www.weave.works/blog/introduction-to-prometheus-monitoring/> [Accessed 9 August 2018].

Taylor, T., 2017, *TOP 10 Open-Source Application Monitoring Tools*. [Online]

Available at: <http://techgenix.com/open-source-application-monitoring-tools/> [Accessed 8 August 2018].

Vunvulea, R., 2017. *Tips and Trick to prepare a demo for Azure Time Series Insights*. [Online]

Available at: <http://vunvulearadu.blogspot.com/2017/12/tips-and-tricks-to-prepare-demo-with.html> [Accessed 12 July 2018].

Watson IoT Platform, n.d. [Online]

Available at: <https://www.ibm.com/cloud/watson-iot-platform> [Accessed 8 July 2018].

BCS (Hons) Computer Science

Faculty of Information and Communication Technology (Kampar Campus), UTAR.

Bibliography

Wikimedia n.d. *Grafana Logo*. [Online]

Available at: https://commons.wikimedia.org/wiki/File:Grafana_logo.png

[Accessed 10 August 2018].

Yigal, A., 2018. *Grafana vs Kibana: The Key Differences to Know*. [Online]

Available at: <https://logz.io/blog/grafana-vs-kibana/>

[Accessed 10 August 2018].

A Data Analytic Module to Extend Grafana Functionality

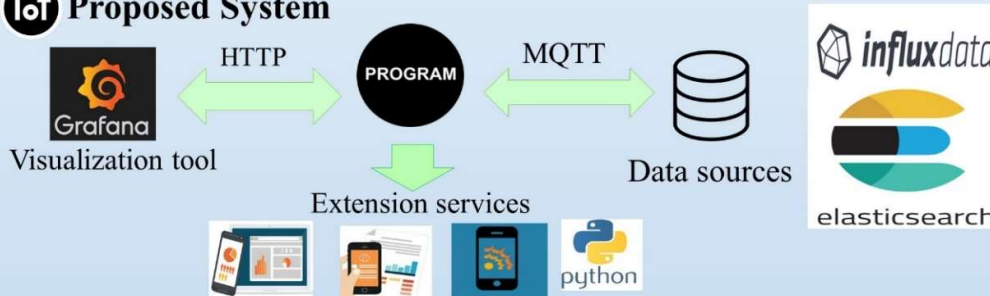
IoT Introduction

- In this data driven era, there is a need for a versatile platform to visualize and analyze the event data.
- Grafana is a free open source, visualization and monitoring tool. Grafana is a well recommended data analytic platform because it is feature rich and various data source.

IoT Problem Statement / Motivation

- Grafana unable to handle and perform custom query function as its function depend on data source
- Grafana cannot visualize the data because perform custom function
- New extension service or application created cannot plugin into the system and use directly.

IoT Proposed System



- Develop a program seat the middle between the Grafana and data source to redirect the custom query and process by other service and solve the custom query function.
- Grafana able to visualize custom function data as Grafana can function independently from data source.
- Extend Grafana Functionality, new extension services or applications can plugin to the system though this module.

IoT Contribution

This project extend the functionality of Grafana. In future Grafana Analytic system can enhance through this module by :

1. Add more extension services
2. Provide access control , improve security
3. Enable system to perform load balancing function, improve performance
4. Increase system availability high fault tolerance and more robust

IoT Conclusion

- IoT analytic platform is important for every individual to gain insight from the data.
- From this project, Grafana able to solve data analytic solution in a very low-cost, high flexibility and less complexity
- Support various kind of data source, work independently from the data source



BCS (HONS) Computer Science
Faculty of Information and Communication
Technology (Perak Campus), UTAR.

Project Developer : Gan Kian Yong
Project Supervisor : Dr Ooi Boon Yaik

Appendix B Turnitin Result

feedback studio | Gan kianyong | A Data Analytic Module To Extend Grafana Functionality

Chapter 1 Introduction

1.1 Project Background

Internet of things analytic is an important application of data analysis and a procedures to explore a huge volume of raw data which collect and capture from those IoT sensor continuously. The cloud end of the IoT consist of three important segment, which include data visualization, storage and the analytic platform. All the raw data capture from the sensor are meaningless and valueless for everyone, it require some analysis and visualization tool to interpret in order to understand the information hiding in the data.

Data visualization is a term that describing the ability to help people understand and explain the large amount of complex data by putting it in a graph,

Match Overview

0%

1	Submitted to Universiti ... <small>Student Paper</small>	<1%
2	repository.up.ac.za <small>Internet Source</small>	<1%
3	magerewardpoints.com <small>Internet Source</small>	<1%

Page: 1 of 49 | Word Count: 8913 | Text-only Report | High Resolution On

Turnitin Originality Report

Document Viewer

Processed on: 06-Apr-2019 00:24 +08
 ID: 1106608352
 Word Count: 8913
 Submitted: 2

A Data Analytic Module To Extend Grafana Func... By Gan kianyong

Similarity Index	Similarity by Source
< 1%	Internet Sources: 0% Publications: 0% Student Papers: 0%

exclude quoted exclude bibliography exclude small matches download print mode: quickview (classic) report
<p><1% match (Internet from 06-Feb-2019) https://repository.up.ac.za/bitstream/handle/2263/45904/Slabbert_Evaluating_2015.pdf?isAllowed=y&sequence=1</p>
<p><1% match (student papers from 11-Apr-2016) Submitted to Universiti Teknologi Petronas on 2016-04-11</p>
<p><1% match (Internet from 01-Apr-2019) http://magerewardpoints.com</p>

Chapter 1 Introduction 1.1 Project Background Internet of things analytic is an important application of data analysis and a procedures to explore a huge volume of raw data which collect and capture from those IoT sensor continuously. The cloud end of the IoT consist of three important segment, which include data visualization, storage and the analytic platform. All the raw data capture from the sensor are meaningless and valueless for everyone, it require some analysis and visualization tool to interpret in order to understand the information hiding in the data. Data visualization is a term that describing the ability to help people understand and explain the large amount of complex data by putting it in a graph, charts or any visual context. (Pittenturf n.d.) Data visualization place an important role in data analysis as it make complex data more understandable and usable. People can understand the information form the data easier and comprehend the tendency by looking at the trend pattern from the graph. Visualize and interpret the complex data help business leader understand information and provide the ability easily drill drop into data identify their weakness in order to improve their business grow. Today's data visualization has become standard for recent business intelligence to support better business decision making and get significant competitive advantage over their business competitor. (Rouse, 2017) Data storage is describing recording data into a storage medium, ensuring access and update managed of data easily and effectively. IoT sensor device always have limited storage capability, time-base data capture from the IoT sensor device will send to the cloud however high volume of data that stored in the cloud will slow down the process of accessing the data. (Gerber, 2017) Time series database is build and design specifically for time-base data storage, it has high availability, support concurrent read and write process and allow very high transaction volume of time series data, query performance can be improve by configure the index to optimize data access. Analytic platform help to perform operation on data analytic, it is a solution design to meet the demand of users or businessman who have large data-drive companies. It provide detail analysis in contextual form from all the data stored though descriptive, predictive and prescriptive analytic. Descriptive analytic to gain insight into the past where it allow us to observe what behaviours happened from the past and understand how it will affect the result and outcome in the future while the predictive analytic make use of some statistical model and various algorithm to identify the relationship among different data set to predict things might happen in future. From the predictive analytics, it provide companies with the actionable insights base on

A Data Analytic Module To Extend Grafana Functionality

ORIGINALITY REPORT

0 %	0 %	0 %	0 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Teknologi Petronas Student Paper	<1 %
2	repository.up.ac.za Internet Source	<1 %
3	magerewardpoints.com Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

Appendix C Supervisor's Comments on Originality Report

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Gan Kian Yong
ID Number(s)	1503490
Programme / Course	Bachelor of Computer Science (HONS)
Title of Final Year Project	A Data Analytic Module to Extend Grafana Functionality

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: _____ % Similarity by source Internet Sources: _____ % Publications: _____ % Student Papers: _____ %	
Number of individual sources listed of more than 3% similarity: _____	
Parameters of originality required and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above

Signature of Supervisor

Signature of Co-Supervisor

Name: _____

Name: _____

Date: _____

Date: _____

Appendix D Checklist for FYP2 Thesis Submission



UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORMATION & COMMUNICATION
TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	150ACB3490
Student Name	Gan Kian Yong
Supervisor Name	Ooi Boon Yaik

TICK (√)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Cover
	Signed Report Status Declaration Form
	Title Page
	Signed form of the Declaration of Originality
	Acknowledgement
	Abstract
	Table of Contents
	List of Figures (if applicable)
	List of Tables (if applicable)
	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
	Chapters / Content
	Bibliography (or References)
	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
	Poster
	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

<p>I, the author, have checked and confirmed all the items listed in the table are included in my report.</p> <p>_____</p> <p>(Signature of Student)</p> <p>Date:</p>	<p>Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction.</p> <p>_____</p> <p>(Signature of Supervisor)</p> <p>Date:</p>
--	--

