# 

**REPORT**

**ON**

**INDUSTRIAL ATTACHMENT**

**WITH**

**MICRON SEMICONDUCTOR ASIA PTE LTD**

**PREPARED BY: HU XIAOXIANG**

**U1521319A**

**EEE**

# Abstract

This report consists of three parts, which are the introduction of the big data techniques, the detail on tester variance project and working flow of a web UI design. As a member of data science team in Micron Semiconductor Backend (MSB), I involved in two different projects during my internship period: Tester Variance Analysis (Individual) and Auto-Diagnostic Tool Development (Group). Both projects enhance my programming skills and communication skills extremely. In this report, my key points will mainly focus on the involved techniques in the project and the relevant documentation.

# Acknowledgement

I would like to express my special thanks of gratitude to my mentor Junchi Mao as well as my supervisor Kian Leong Lim who gave me the golden opportunity to do this wonderful internship project in MSB data science team. They also helped me in doing a lot of teamwork and communication with other department and I came to know about so many new things. I am really thankful to them.

Contents

[Abstract 2](#_Toc483487185)

[Acknowledgement 3](#_Toc483487186)

[Introduction of Big Data and relevant analytic tools 6](#_Toc483487187)

[Background 6](#_Toc483487188)

[Hadoop Platform 6](#_Toc483487189)

[HDFS Architecture 7](#_Toc483487190)

[MapReduce Method 7](#_Toc483487191)

[Apache Hive 8](#_Toc483487192)

[Apache Pig 8](#_Toc483487193)

[Spark 8](#_Toc483487194)

[Processes of Data Analysis 9](#_Toc483487195)

[Project 1: Tester Variance Analysis 10](#_Toc483487196)

[Background 10](#_Toc483487197)

[Objective 10](#_Toc483487198)

[Technique Detail 11](#_Toc483487199)

[Data Cleaning 11](#_Toc483487200)

[Automation Working Flowchart 15](#_Toc483487201)

[Data Visualization 16](#_Toc483487202)

[Summary 17](#_Toc483487203)

[Project 2: Development of On-demand Web UI for Auto-diagnostic Tool (ADT) 18](#_Toc483487204)

[Background 18](#_Toc483487205)

[Introduction of ADT 18](#_Toc483487206)

[Web Framework 18](#_Toc483487207)

[Model–View–Controller Architecture 18](#_Toc483487208)

[Django 19](#_Toc483487209)

[Channel and WebSocket 19](#_Toc483487210)

[Development Procedure 20](#_Toc483487211)

[Identification of Users’ Requirement 20](#_Toc483487212)

[Framework Setup 20](#_Toc483487213)

[Work Flow 21](#_Toc483487214)

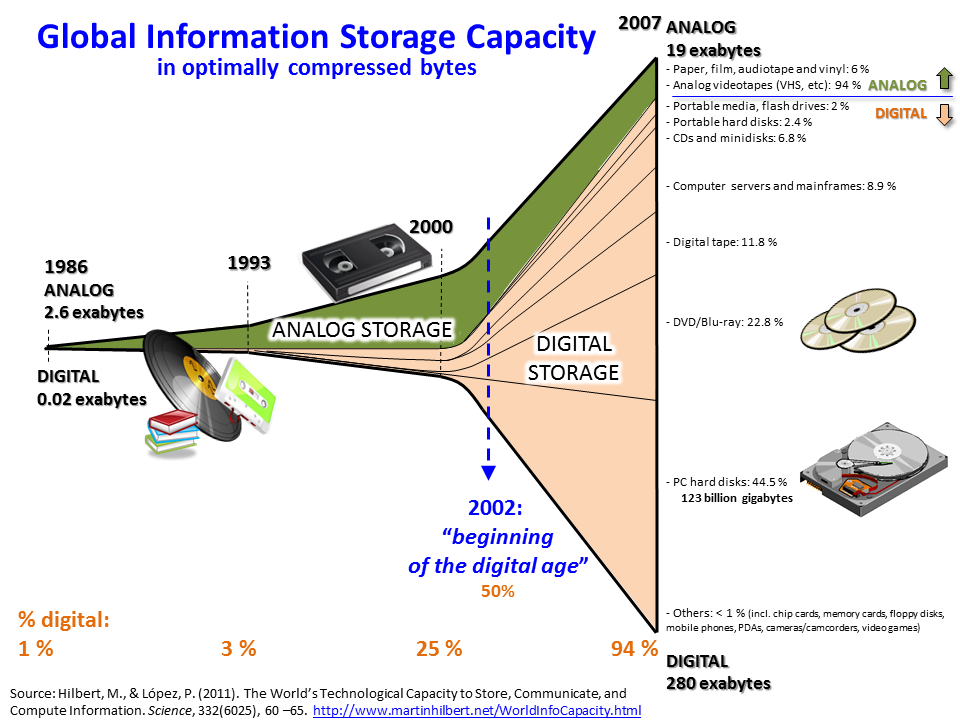
[Conclusion 23](#_Toc483487215)

[References 24](#_Toc483487216)

# Introduction of Big Data and relevant analytic tools

## Background

Data which is generated and collected by humanity is rapidly increasing since the last few decades. According to some researches, the data generated in the last few years has been more than the sum of data generated in the last 20 centuries. To handle this level of amount of data, we need to develop a new kind of tools to overcome the storage, transfer and process challenges faced by traditional data analytic tools. The term ‘Big Data’ steps closely to people’s life within this special scene of techniques development.

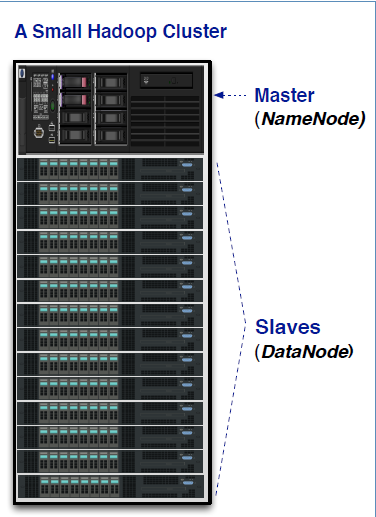
Generally, Big Data is a term for data sets that are so large or complex that traditional data processing application software is inadequate to deal with them. Challenges include storage, analysis, search, transfer, visualization, querying and updating.

## Hadoop Platform

Apache Hadoop is an open-source software framework used for distributed storage and processing of big datasets using the MapReduce programming model. The core Hadoop consists of two main components: Storage-the Hadoop Distributed File System (HDFS) and Processing-MapReduce Method.

Data processing jobs in Hadoop are broken into individual tasks. Each task takes a small amount of data as input and thousands of tasks often run in parallel at the same time. That is the reason why Big Data platform has the capability to deal with many times of amount of data more than traditional tools and still has a relatively acceptable processing time. Moreover, the Hadoop platform is designed to handle failure automatically. The concept behind it is the hardware failure is common and inevitable. So if a node fails during processing, its tasks are rescheduled elsewhere automatically.

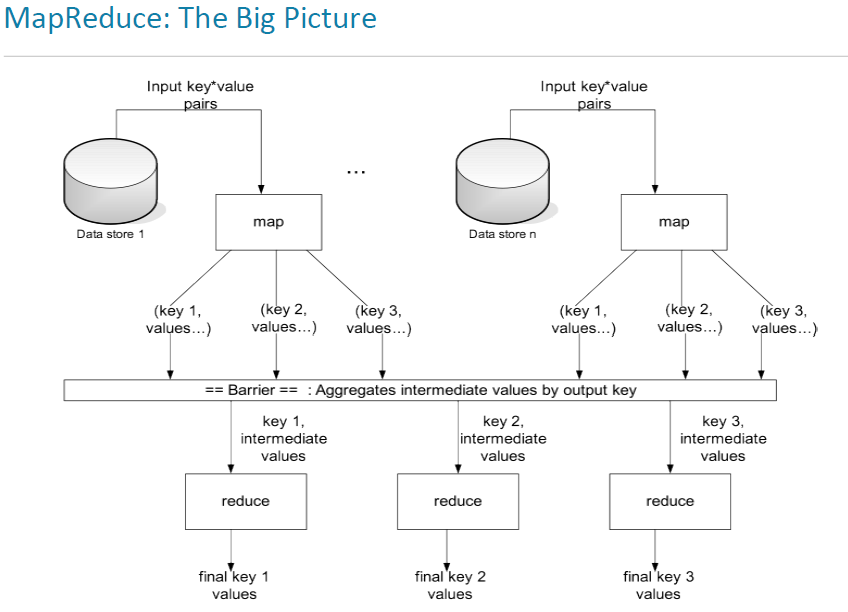
### HDFS Architecture

HDFS has a master/slave architecture. HDFS master daemon, which is called NameNode, is used to manage namespace and metadata and meanwhile monitor slave nodes. The slave daemon also known as the DataNode, is charged of reading and writing the actual data.

### MapReduce Method

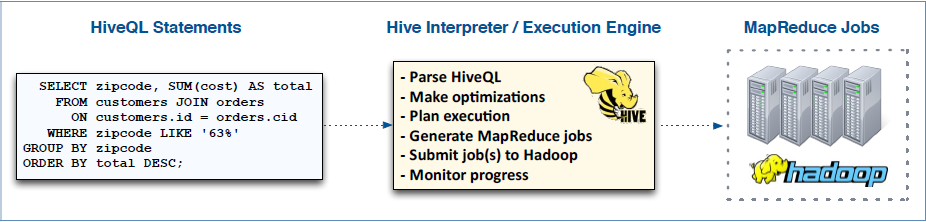
MapReduce, which is popularized by Google in 2004, is a programming model for simple data processing on large clusters. The three main benefits of MapReduce are simplicity, flexibility and scalability.

MapReduce consists of 2 functions: Map and Reduce. The output from map becomes the input to reduce. The map function always runs first to filter, transform, or parse data, and the reduce function is optional.



### Apache Hive

Apache Hive is a data warehouse software project built on top of Apache Hadoop for providing data summarization, query, and analysis. Hive gives an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop.



### Apache Pig

Apache Pig is a high-level platform for creating programs that run on Apache Hadoop. The language for this platform is called Pig Latin. Pig can execute its Hadoop jobs in MapReduce, Apache Tez, or Apache Spark. Pig Latin abstracts the programming from the Java MapReduce idiom into a notation which makes MapReduce programming high level, similar to that of SQL for relational database management systems.

## Spark

Apache Spark is an open-source cluster-computing framework. Spark provides an interface for programming entire clusters with implicit data parallelism and fault-tolerance.

Apache Spark provides programmers with an application programming interface centered on a data structure called the resilient distributed dataset (RDD), a read-only multiset of data items distributed over a cluster of machines that are maintained in a fault-tolerant way. It was developed in response to limitations in the MapReduce cluster computing paradigm. Spark's RDDs function as a working set for distributed programs that offers a restricted form of distributed shared memory.

Furthermore, the processing speed of Spark is usually 10 to 100 times faster than Hadoop, e.g. Hive. The Spark Core is the foundation of the overall Spark project. It provides distributed task dispatching, scheduling and basic I/O functionalities. In addition, Spark supports other application programming interfaces like Java and Python, which provides a good extensibility on its functions.

## Processes of Data Analysis

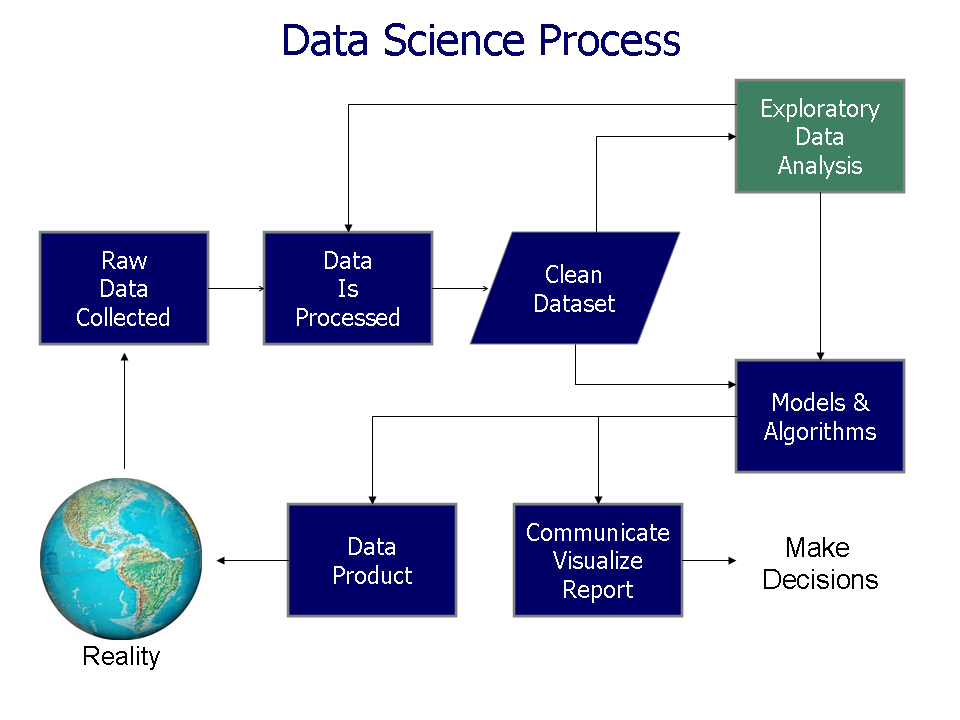
Generally, the processes of data analysis can be categorized into 4 steps: data collection, data pre-processing, data cleaning and data analysis.

Firstly, data is collected from a variety of sources. The requirements may be communicated by analysts to custodians of the data, such as information technology personnel within an organization. The data may also be collected from sensors in the environment, such as traffic cameras, satellites, recording devices, etc.

The next step is to pre-process the data in a table with structured format and then store them into the Big Data Storage, e.g. HDFS.

The major job I did here is mainly about the data cleaning and data analysis. It is common to know that data may be incomplete, contain duplicates, or contain errors even after the pre-processing step. Thus a cleaning step is very necessary to prevent and correct these errors before feeding data into modeling. Typically, data cleaning includes the following jobs: record matching, identifying inaccuracy of data, overall quality of existing data, deduplication and column segmentation.

The final step is data analysis. In this procedure, modeling and algorithms may be applied to the data to identify relationships among the variables. Data visualization may also be used to obtain additional insight.



# Project 1: Tester Variance Analysis

## Background

Tester Variance is the difference between the input and output unit from a tester machine. For example, there are 1000 units fed into tester and for some unknown reason the output unit number is only 900. This kind of problem is called tester variance.

After consulting the process team, some concluded reasons are listed here:

1. Double-Stack Issue

The double-stack issue happens when the tester probe occasionally laid two IC chips on the same tray. This issue is quality related because the top one may not be tested. This type of issue always requires rescreen procedure to be applied again to insure the good quality.

1. Pick & Place Issue

This issue happens when the probe sometimes drop components on the way. This is not a quality-related problem.

1. Stuck in the tray Issue

Sometimes the IC chip may be stuck on somewhere between the tray and cause this issue. This is also not a quality-related problem.

The major problem caused by this tester variance problem is that it takes extra time for recheck every tray, so to make sure no any tester variance case is caused by double-stack issue. Usually this step is very time consuming. If we can find a way to reduce the overall tester variance cases, we can improve the production efficiency.

## Objective

This project has 2 objectives:

* Automate tester variance data collection for easier troubleshooting
* Provide tester variance data visibility for intuitive data analysis

## Technique Detail

### Data Cleaning

To automate the entire data collection process, I need to select discrete data from Hadoop and then combine them to generate a new table. For data cleaning procedure, it is very important to understand the relationship among different lot attributes.

**Step\_job\_oid**

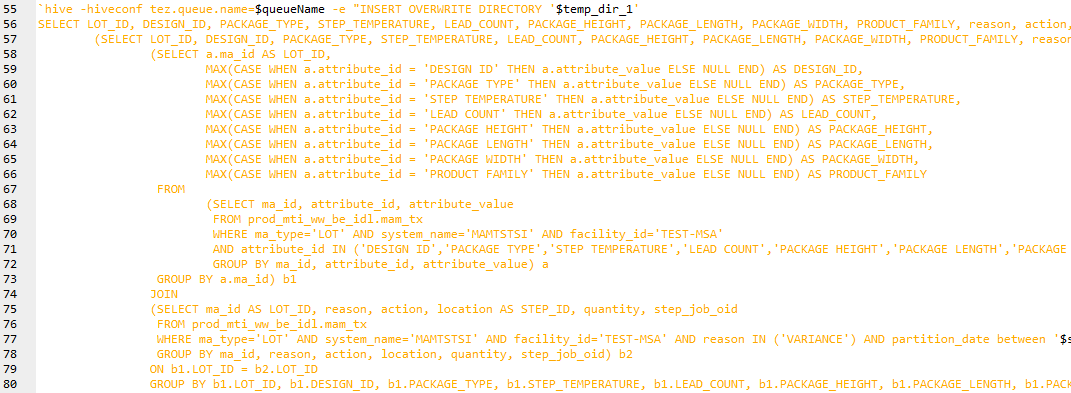
**Lot\_id**

**Total Lot quantity**

**Tester ID**

**Other key information**

The actual query statement is written in Perl script.

To embedded Hive-QL in Perl script:

The Hive Query Language (HiveQL) is a query language for Hive to process and analyze structured data in a Metastore.

The most common used clauses in HiveQL are SELECT…FROM, WHERE, JOIN, GROUP BY.

There are different types of joins given as follows:

* JOIN
* LEFT OUTER JOIN
* RIGHT OUTER JOIN
* FULL OUTER JOIN

Each of these type of join is used for different purpose and different condition.

JOIN is same as OUTER JOIN in SQL. The joined table contains either all the records from both the tables, or fills in NULL values for missing matches on either side.

The HiveQL LEFT OUTER JOIN returns all the rows from the left table, even if there are no matches in the right table. This means, if the ON clause matches 0 (zero) records in the right table, the JOIN still returns a row in the result, but with NULL in each column from the right table.

The HiveQL RIGHT OUTER JOIN returns all the rows from the right table, even if there are no matches in the left table. If the ON clause matches 0 (zero) records in the left table, the JOIN still returns a row in the result, but with NULL in each column from the left table.

Required relationship table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **LOT\_ID** | **STEP\_ID** | **DESIGN\_ID** | **PACKAGE\_TYPE** | **PACKAGE\_HEIGHT** | **PACKAGE\_LENGTH** | **PACKAGE\_WIDTH** | **PRODUCT\_FAMILY** | **TESTER\_ID** | **LOT\_QTY** | **reason** | **quantity** | **date\_time\_local** | **ww** |
| 1234567.01 | HOT SORT | M69A | VFBGA | 1.000 | 11.000 | 9.000 | NAND FLASH | TAW-0020 | 11 | VARIANCE | -1 | 2016-08-13 05:12:43.21 | 201611 |

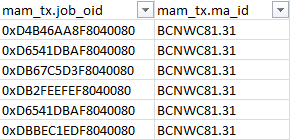
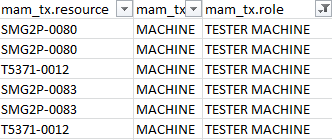
The final relationship table looks like the one on above. To generate this table, we need to combine the different fields from other 4 tables.

Table 1



**condition: reason= VARIANCE**

Table 2



**condition: reason= step\_job\_oid = job\_oid**

Table 3

**condition: Attribute\_id = CURRENT QTY and step\_job\_oid = job\_oid**

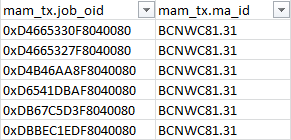
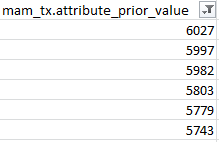
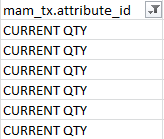
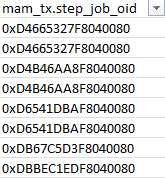
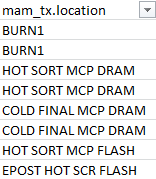
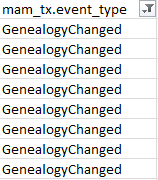


Table 4



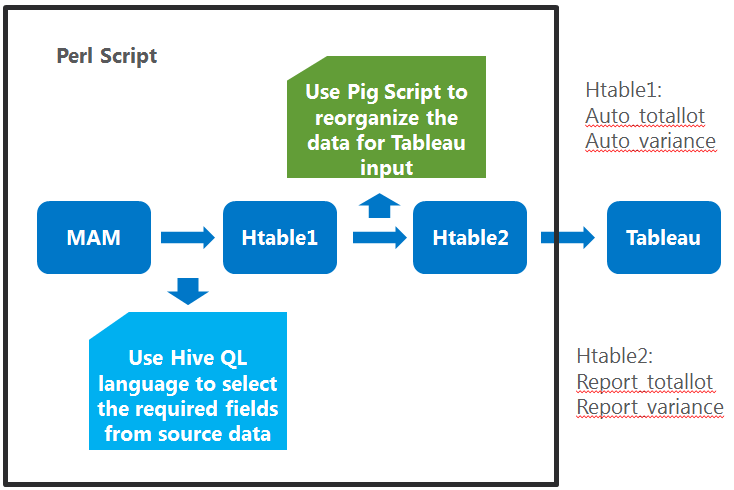
**condition: event\_type = GenealogyChanged and step\_job\_oid = job\_oid**

**: Join condition**

**: Target data**

From the table above, we can find all of these different attributes, e.g. location, resource, ma\_id, are related to job\_oid or step\_job\_oid. This is the key to link all these fields together.

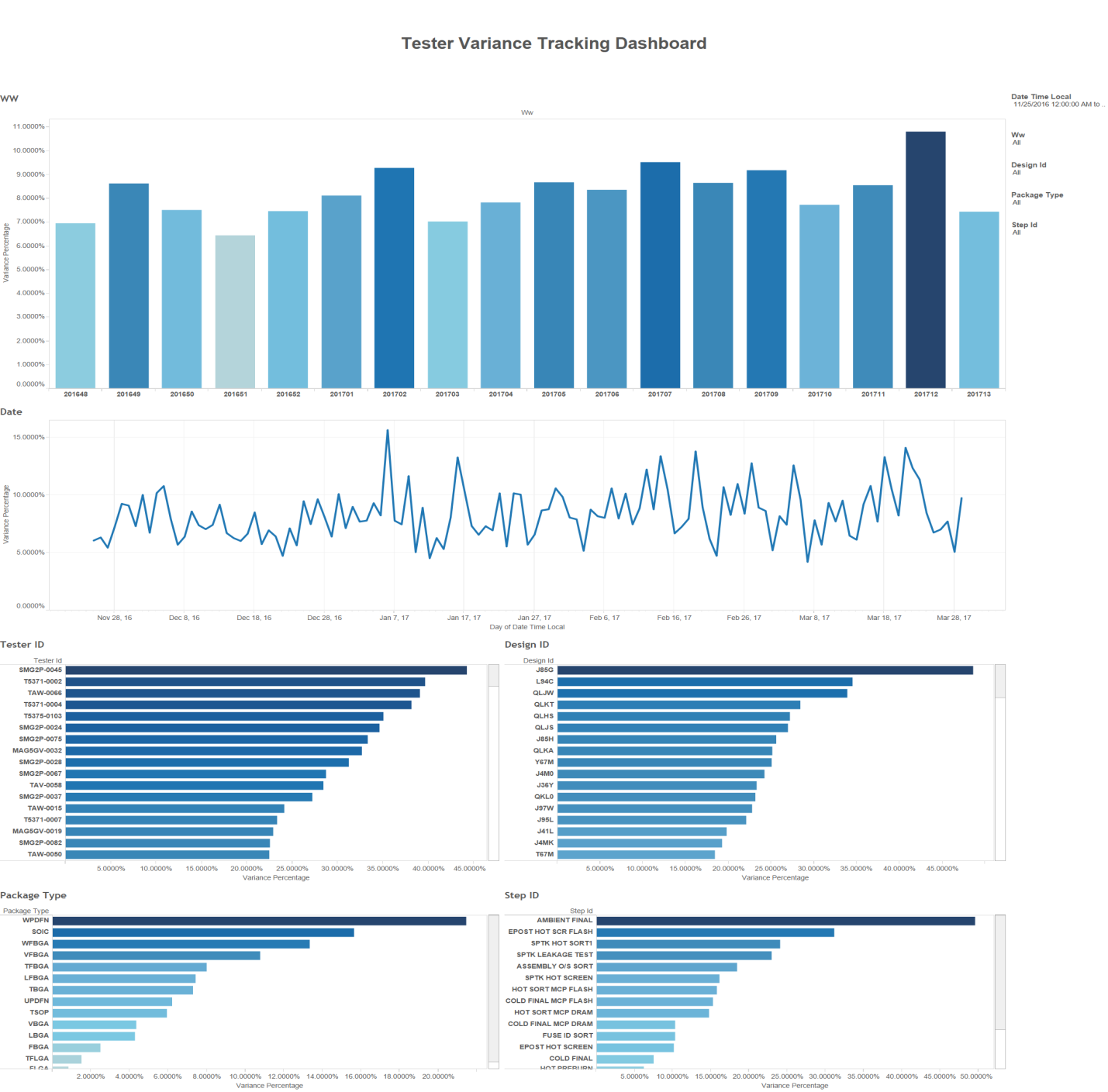
### Automation Working Flowchart



This flowchart shows how the automation works from the beginning to the end. Firstly, I use HiveQL to construct a cleaned structured table in Hive called Htable1. This table will refresh every day to ingest the updated data from MAM. Then, on every Thursday, I use Pig to collect all last 1-year long data from Htable1 and reorganize it into Htable2. Tableau will read the data from Htable2 directly.

### Data Visualization

Tableau is a Business Intelligence tool for visually analysing the data. Users can create and distribute interactive and shareable dashboards which depict the trends, variations and density of the data in form of graphs and charts. Tableau can connect to files, relational and Big data sources to acquire and process data. The software allows data blending and real time collaboration, which makes it very unique. It is used by businesses, academic researchers and many governments to do visual data analysis. In this project I use Tableau Dashboard as the data visualization tool.



Variance Percentage = Total Number of Lot with Tester Variance / Total Number of Lot

The Variance Percentage is the index to indicate how much percentage of lot with tester variance can be within the total number of lot. In the dashboard, user can also filter the table by setting different condition. For example, if filtering by Tester ID, it is easily to notice that the top item has a unusual high variance percentage compared to others. This indicate there may be some potential problems with this tester machine.

## Summary

According to the feedback from the process team, the automation work of tester variance data collection will save 1 to 1.5 work hours per week. In addition, data visibility using Tableau dashboard can provide an easier way for troubleshooting tester variance problem

# Project 2: Development of On-demand Web UI for Auto-diagnostic Tool (ADT)

## Background

### Introduction of ADT

* Develop auto-diagnostics tool for 2DID products
  + looking for machine, operator and material commonality
* Provide root cause(s) analysis of an Out of Control (OOC) case
  + Base on AOST SBL result
* Provide raw data for engineers to mitigate root cause

### Web Framework

A web framework is a software framework that is designed to support the development of web applications including web services, web resources, and web APIs. Web frameworks, provide a standard way to build and deploy web applications. Web frameworks aim to automate the overhead associated with common activities performed in web development.

### Model–View–Controller Architecture

Model–view–controller (MVC) is a software architectural pattern for implementing user interfaces on computers. It divides a given application into three interconnected parts in order to separate internal representations of information from the ways that information is presented to and accepted from the user.

* The *model* is the central component of the pattern. It expresses the application's behavior in terms of the problem domain, independent of the user interface. It directly manages the data, logic and rules of the application.
* A *view* can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.
* The third part, the *controller*, accepts input and converts it to commands for the model or view.

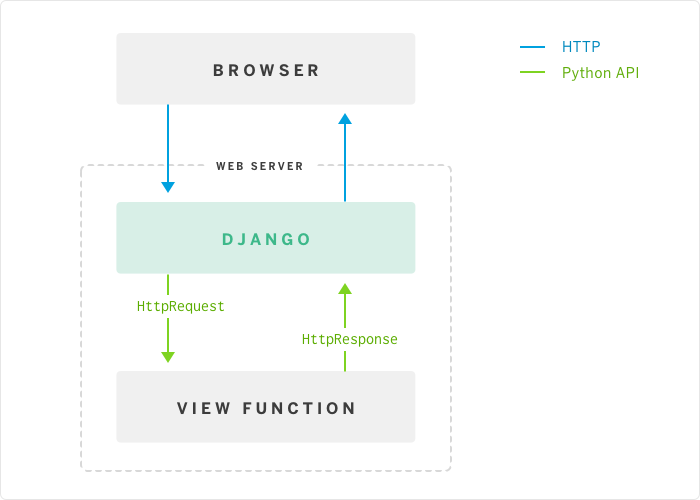
### Django

Django is an open-source web framework, written in Python, which follows the MVC architectural pattern.

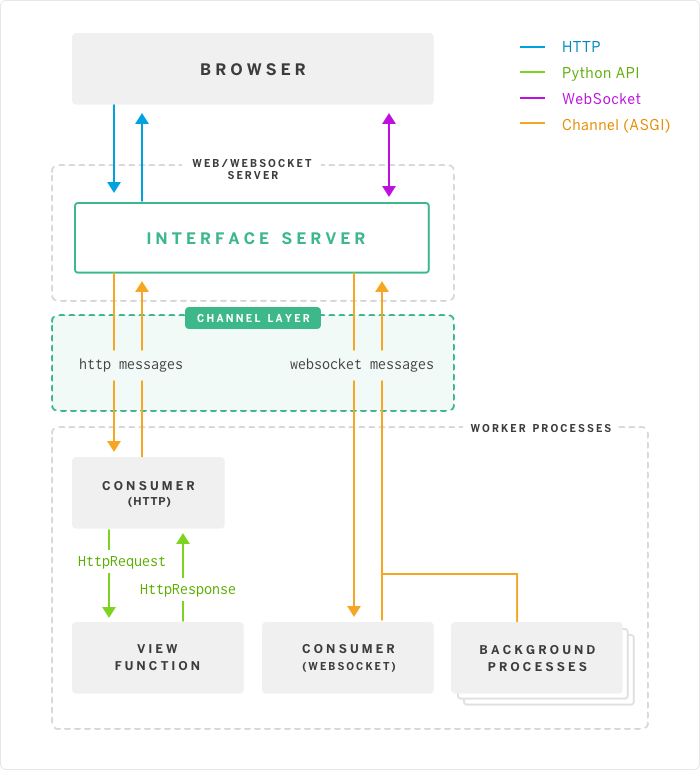
To install Django: pip install django

The most frequently used functions in Django are its layers, including its model layer, view layer and template layer. The form also has template for quick deployment.

### Channel and WebSocket

Channels is a project to make Django able to handle more than just plain HTTP requests, including WebSockets and HTTP2, as well as the ability to run code after a response has been sent for things like thumbnailing or background calculation.

WebSocket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection.

WebSocket is designed to be implemented in web browsers and web servers, but it can be used by any client or server application. The WebSocket Protocol is an independent TCP-based protocol. Its only relationship to HTTP is that its handshake is interpreted by HTTP servers as an Upgrade request. The WebSocket protocol enables interaction between a browser and a web server with lower overheads, facilitating real-time data transfer from and to the server. This is made possible by providing a standardized way for the server to send content to the browser without being solicited by the client, and allowing for messages to be passed back and forth while keeping the connection open. In this way, a two-way (bi-directional) ongoing conversation can take place between a browser and the server.

## Development Procedure

### Identification of Users’ Requirement

Input:

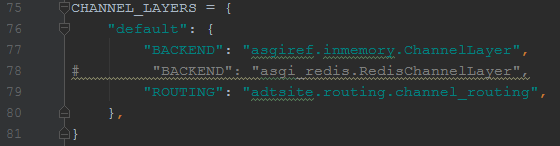
* Lot list including both good and bad lot
* Date range to specify analysis period

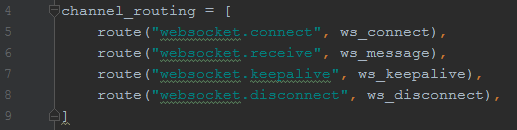
Output:

* Random Forest and Kruskal-Wallis analysis results
* Raw data for results verification

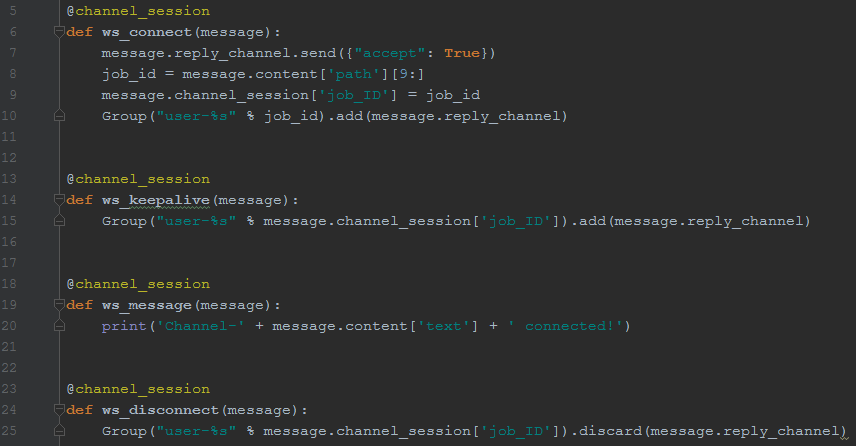
### Framework Setup

Use channel layer:

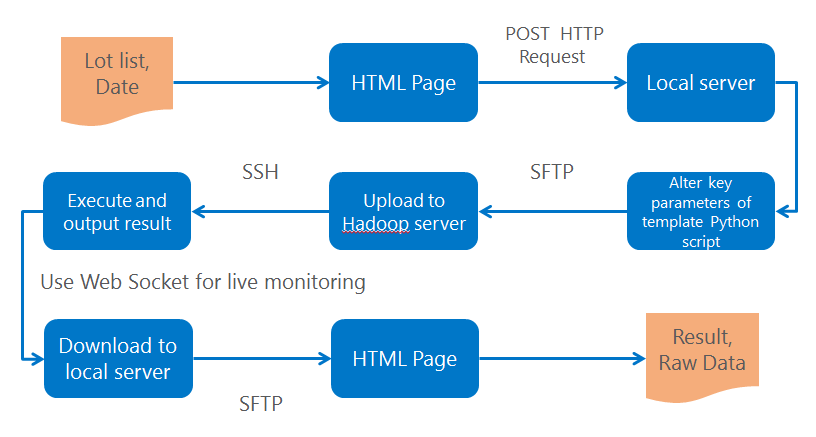


Route incoming message:

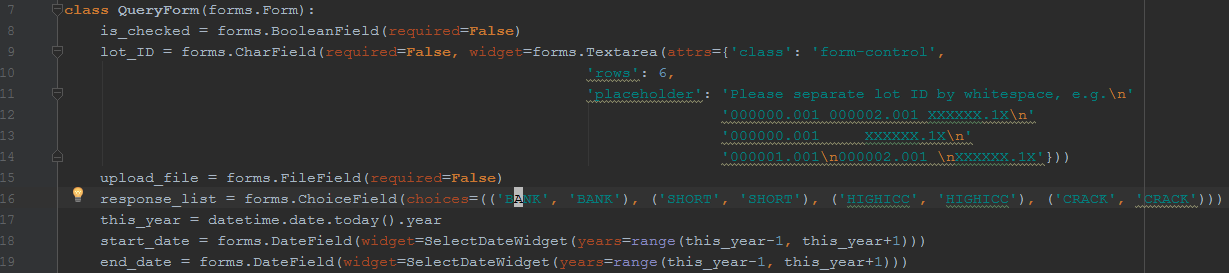
Consume message:



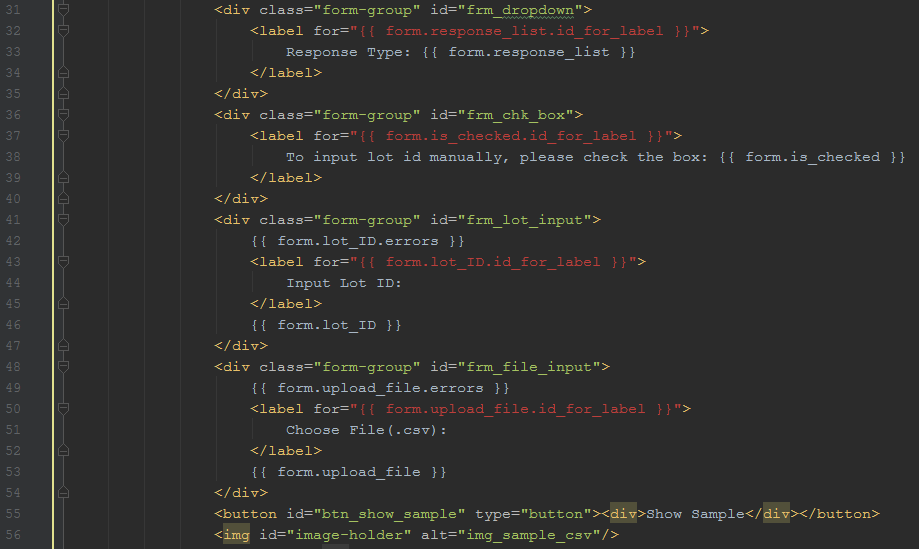
### Work Flow



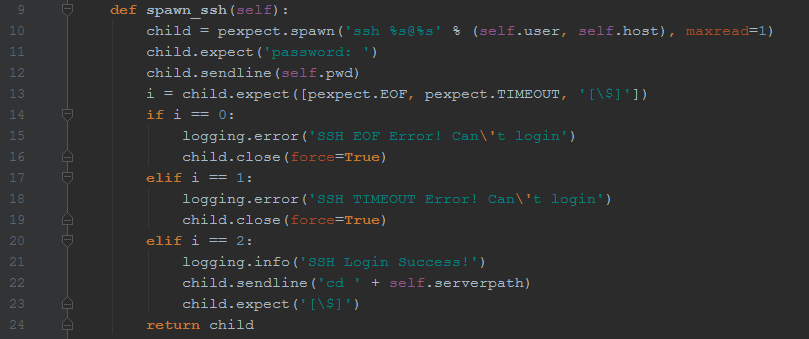
To render HTML page for input fields, we need to use Django form template.

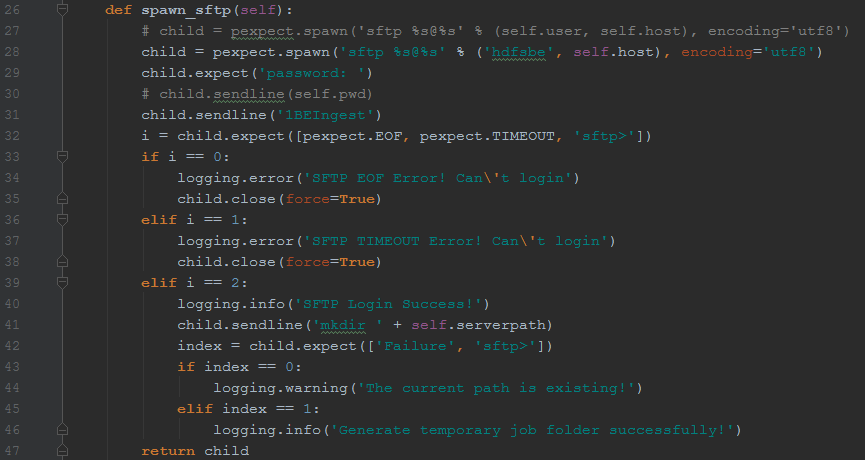


Manually rendering on HTML page:



Use pexpect to interact with SSH and SFTP:





# Conclusion

Overall, this internship experience gives me a thorough exposure on the industrial environment. During the internship period, each of the assigned project let me learn new technique skills as well as realize the importance of teamwork.

As conclusion, the first project tester variance analysis provides me an opportunity to have a glimpse of the production line in a semiconductor company like Micron. It let me know how different departments collaborate with each other to insure the quality and cycle time of the production. Meanwhile, the second project let me see the chance to apply IT technologies in this industry, such as web design.

Since I am working in the data science team in Micron, this internship experience indeed inspire me to pursue data science as my future academic career. It is also very excited to know how data science can make a change on the production area.

# References

<https://en.wikipedia.org/>

<https://blog.heroku.com/in_deep_with_django_channels_the_future_of_real_time_apps_in_django>

<https://docs.djangoproject.com/en/1.11/>

Cloudera Developer Training for Apache Hadoop

Cloudera Data Analyst Training