Data Analytics And Application Life Cycle Development Using Gulp Task Runners

Submitted By

ROTALIWALA KIRTIKUMAR BHARATKUMAR

16MCEC17

DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2018

Data Analytics And Application Life Cycle Development Using Gulp Task Runners

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

ROTALIWALA KIRTIKUMAR BHARATKUMAR

(16MCEC17)

Guided By

Assistant Professor Vishal Parikh

DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2018

Certificate

This is to certify that the major project entitled **"Data Analytics And Application Life Cycle Development Using Gulp Task Runners"** submitted by **ROTALIWALA KIRTIKUMAR BHARATKUMAR (16MCEC17)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-II, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Assistant Professor Vishal Parikh
Guide & Assistant Professor,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Priyanka Sharma
Associate Professor,
Coordinator M.Tech - CSE (Specialization)
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **ROTALIWALA KIRTIKUMAR BHARATKUMAR**, **16MCEC17**, give undertaking that the Major Project entitled "**Data Analytics And Application Life Cycle Development Using Gulp Task Runners**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Assistant Professor Vishal Parikh

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Assistant Professor Vishal Parikh**, Associate Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- ROTALIWALA KIRTIKUMAR BHARATKUMAR
16MCEC17

Abstract

In application life cycle model, there are many repetitive tasks to be performed right from development stage up to the application usage stage. Each task involves human effort and time though being routine in nature. This then impacts the cost and time for application life cycle. It is must be noted that application development cost also includes the testing phase cost in the application life cycle. It is very important for any application life cycle to fine tune this operation model and thereby lower the cost. In other use cases, when the application is deployed for usage, the application may involve network related operations whereby it may interact with local controller in a subnet or a remote controller. In such use cases, the response time of application while it accesses the controller plays an important role in quality of application. The network traffic problem over the internet is a formidable problem in the world of networked devices where different devices such as IoT devices have exploded in the network.Gulp task runner helps in the automation of all repetitive tasks that can be usually employed systematically in application life cycle from development to application usage stage. It is accepted that such automation can save significant human effort and eliminate errors and thereby reduce testing costs.We believe that with automated task runners used in application life cycle, we can demonstrate savings in effort and cost and thus focus our effort more to analyze the application usage involving data traffic, failure analysis and accordingly adapt with corrective actions such as improving the throughput of the whole system.

Abbreviations

JS	JavaScript.
GUI	Graphical User Interface.
TDD	Test Driven Development (programming methodology).
DOM	Document Object Model.
CLI	Command Line Interface.
IDE	Integrated Development Environment.
IoT	Internet of Things.
PLC	Programmable Logic Controller.
CPE	Customer Premise Equipment.
ACS	Auto Configuration Server.
SOAP	Simple Object Access Protocol.
RPC	Remote Procedure call.
SNMP	Simple Network Management Protocol.
Data Object	Object set for management of a CPE.
TR	Technical Report.
CWMP	CPE WAN Management Protocol.
SSID	Service Set Identifier.
RSSI	Received Signal Strength Indicator.
SQL	structured query language.

Contents

List of Figures

Chapter 1

Introduction

In application life cycle model, there are many tasks which we must perform manually. So, there is extra human effort are invest and extra time is also investing in that. So, cost and time duration of application life cycle is also increase. We already know majority of application development cost is used in testing phase of application life cycle. Also, the response time of accessing the application plays important role in quality of application or software. So, Optimize and speedy source code of application is also very important for the performance of application. And traffic problem over the network is a very big problem in the world of computer science where different devices of IoT are attached with network.

We are trying to solve the problem with the help of task runners. In this context, task is a set of lines of code, which individually performs some important single job. Means each task is dedicated to one important job of each phase of application life cycle. We define the task runners, task runners are the technology which manages and organize the sequence of execution of tasks for to perform the job in each phase of application life cycle as well as to solve the problem of data traffic and failure over the network. In the market many task runners are used like grunt, gulp, yarn, cake, broccoli, etc. But, only two out of them are used majorly grunt and gulp task runners. But there are some benefits of gulp over the grunt task runners. Gulp task runner is about streams and code-over-configuration not only about configuration file like Grunt task runner. Gulp task runner uses the Node, but developer need not to know about that unlike Grunt task runner. The advantage of Gulps code-over-configuration approach is that you end up

with a cleaner and easier to read task file with greater consistency between tasks. Each Grunt task can be considerably different in setup. Gulp task runner uses the stream, which is very good. Because, when we put the file in stream format then we need not worry about temporary file and folder. But, Grunt task runner uses the temporary file and folder policy to read and write file. Plugins of Gulp task runner is simpler than Grunt task runner. Because, plugin of Gulp task runner do single job. Gulp technology manages, organizes and executes tasks in a proper sequence to achieve execution of job of each phase of application life cycle.

Gulp technology is a task runner, which helps in the automation of all repeatable tasks that are usually run systematically. Automation can save significant human effort and eliminate errors. Gulp technology enables automating repetitive tasks. Here, we are making autonomous environment for application life cycle as well as for to solve the problem of data traffic and failure problem over the network. All the above tasks may be employed in several stages of application usage. Stages of the application may be: development, testing, releasing production code, use of application by end user, interaction of local application with other agent applications in the same device and interaction of the local application with the cloud services. Application considered in this study are GUI applications developed for web based desktop applications and mobile apps that involve the embedded customer premises equipment (CPE) devices.

Task runners may be employed in data analytics stage that usually involve data retrieval, data insertion and/or data update during data analysis. Data analysis may be done either locally in centrally located devices (FOG) or remotely in servers (CLOUD). Cloud computing is for the transferring of computing services servers, networking, analytics and more over the Internet like AWS (Amazon Web Services, Microsoft Azure, GCP (Google Cloud Platform). With automated task runner used in data analysis we can then focus our effort more to understand the behavior of data traffic, failure analysis and according adapt with appropriate action such as improving the throughput of the whole system.

It must be noted that task runners employed in applications can flexibly work with all types of devices (mobile, desktop, etc.) and all kinds of platforms. Gulp task runners are best suited in application development that employs technologies such as JS, HTML, CSS, Angular, Android, iOS based code for development. And when we are in data analytics phase that many device data are store on local controller means FOG. This is also very important that how we are make communication between Customer Premise Equipment or embedded devices and local controller. We also implement the authentication policy between those Customer Premise Equipment or embedded devices and local controller. So, access of Customer Premise Equipment or embedded devices will not be misuse. otherwise, it will make the disaster. Those devices are IoT devices like PLCs, routers, modem, phone, set-top boxes, Television, etc. These types of data also describe the health of devices, which are connected over the network.

We can compare these two task runners Grunt and Gulp based on number of available plugins, number of download in January 2016, number of contributors on Git-hub, number of watchers on Git-hub, number of project likes, execution time of common css file compiling. In figure 1.1 yellow color represent the Grunt task runner and red color represent the Gulp task runner. 3349 more plugins in Grunt task runner than Gulp task runner in January 2016, 268317 more downloads in Grunt task runner than Gulp task runner in January 2016, 100 more number of contributors on Git-hub in Grunt task runner than Gulp task runner in January 2016, 324 more number of watchers on Git-hub in Grunt task runner than Gulp task runner in January 2016, 8791 more number of project likes in Grunt task runner than Gulp task runner in January 2016, Grunt task runner takes more 709ms as compare to Gulp task runners. But, it is just for estimation. This is the source from Git-hub and npm. Here, we can be observing that Grunt task runner has more numbers of available plugins, number of download, number of contributors on Git-hub, number of watchers on Git-hub, number of project likes. Because, Gulp task runner is lately introducing than Grunt task runner. All these values of parameters are increasing day by day in Gulp task runner side, and decreasing in Grunt task runner side. So, we can say that Gulp task runner is good as compare to Grunt task runner.

Grunt Task Runner VS Gulp Task Runner

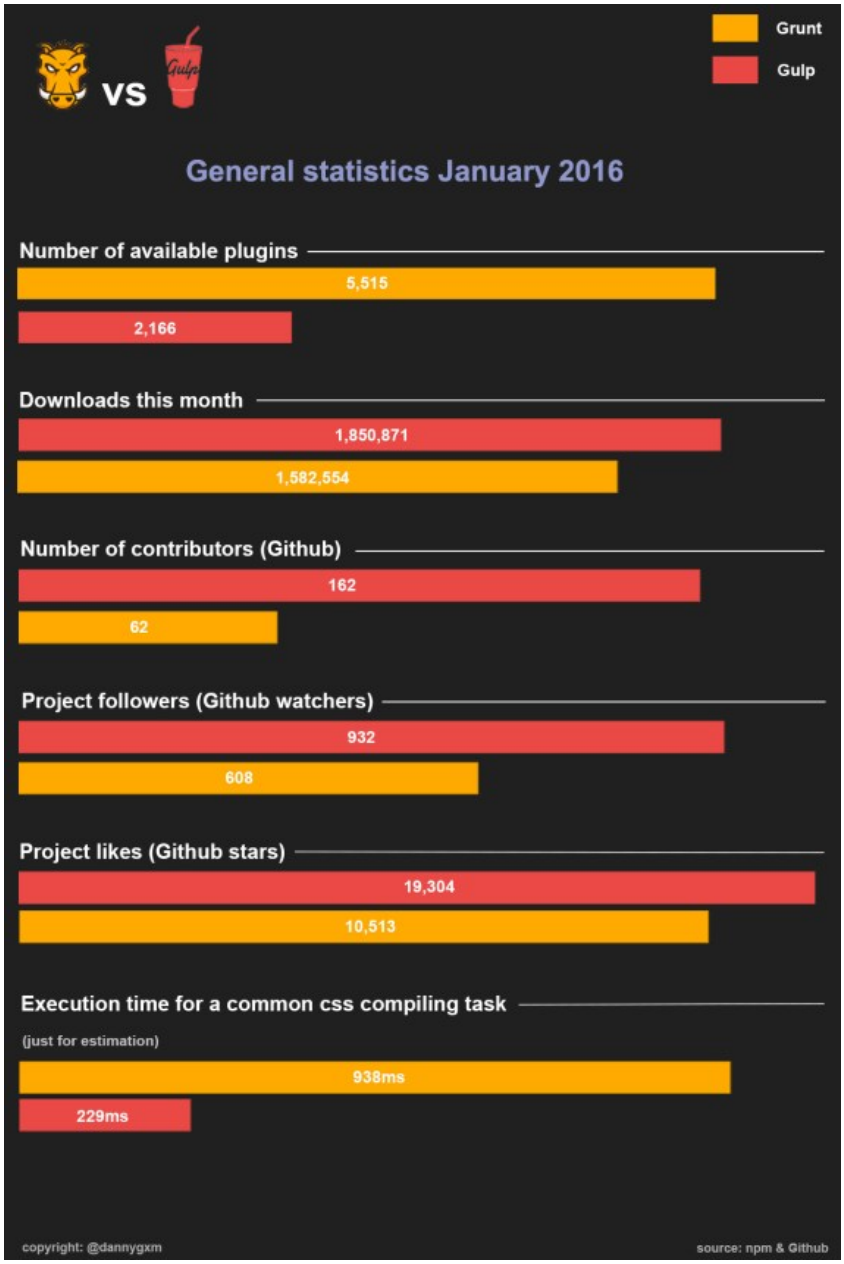

Figure 1.1: Grunt Task Runner VS Gulp Task Runner[1]

Chapter 2

Literature Survey

For to solve above problem, we must use some automation technology, which automate all the rapidly used tasks. So, we can save human effort for those tasks. We are observing task runners for to automate and optimize the source code of application and for automation in unit testing. All type of application likes web application, desktop application, and mobile application and domain of these application are Android, iOS, AngularJs, and those applications are can on any digital devices and on IoT devices like PLCs, Set-Top boxes, Television set, mobile, router, modem etc. And all of this can use automation functionality. And there is another concept of End-to-End testing scenario. Which is also one type of testing scenario, this uses some other type of framework. We have many types of frame work, which will work with karma tool like Mocha, Chai, Jasmine for unit testing modules. But, we are focusing on End-to-End testing scenario then we go with protractor, which is doing same operation in End-to-End testing scenario. Protractor is a framework and it uses Karma tool. So, this is the pair of tool and framework for End-to-End testing scenario, which is use by todays market. And gulp task runner also performs well with this Karma tool and Protractor framework in End-to-End testing scenario. Gulp has some different plug-ins for this End-to-End testing scenario. Also, maintain configuration file which include different attribute and their respective values. In the market, there is many types of task runners are available like Grunt, Gulp, Cake, Yarn, Broccoli, etc. But, we are focusing on Gulp task runners because speed of Gulp task runner is very high than other task runners. Gulp task runner provide many ready-made plugins for each single task, which we want to perform on source of application. Gulp task runner is a Streaming build system.

So, it does not require temporary file or folder system. We are using many types of plugins like minify, uglify, sourcemap, imagemin, concat, reversion, browsify, plumbering, vulcanize, jsHint, encryption. We can also make some new plugins as per our requirements. So, we can directly implement on our projects. Gulp task runners also gives some benefits about dealing with database, because there are some plugins are available for interaction with database like MySql and with MongoDB. And, we can perform database query with Gulp task runner through plugins. When we want do data analytics then, we can use Gulp with MongoDB for unstructured data storage. Gulp task runner also provide the facilities for testing phase of application life cycle. We have many types of frame work, which will work with karma tool like Mocha, Chai, Jasmine, etc. But, as per the usability Jasmine framework provides the good compatibility with Karma and Gulp task runner. And protractor is very good for end to end testing scenario. Where jasmine frame work is appropriate for unit testing scenario. Gulp task runner also provide good compatibility with AWS (Amazon Web Services) through plugins.

Chapter 3

Approach

3.1 Development & unit testing:

Development phase takes use of JS, HTML, CSS, Angular JS, Android, iOS based code which is used with all type device applications like for mobile application, desktop application, etc. And unit testing uses the task runners for applying unit test on module of the project. This is UI phase. This is basically TDD (Test-driven development) testing. Test-driven development is related to the test-first programming concepts of extreme programming. We are using karma as a tool, jasmine as a framework and phantomJS as a browser for displaying the result of TDD. Jasmine is a behavior-driven development framework for testing JavaScript code. It does not depend on any other JavaScript frameworks. It does not require a DOM. And it has a clean, obvious syntax so that you can easily write tests. Benefits of Jasmine is no overheads and no other dependencies. This framework projects the result of test on the command prompt as well as browser. And it is open source. Karma is a tool that enables the running of source code (i.e. JavaScript) against real browsers via the CLI. The fact that it runs against real browsers rather than fakes with a virtual DOM is extremely powerful. The main goal for Karma is to bring a productive testing environment to developers. The environment being one where they dont have to set up loads of configurations, but rather a place where developers can just write the code and get instant feedback from their tests. Because getting quick feedback is what makes you productive and creative. We can test our code on real devices like phones, tablets or on a PhantomJS. Karma can control remotely by CLI or IDE. Karma can use many frameworks like Mocha, Jasmine, Chai, etc.[14] it is easy to debug and it

is also open source. PhantomJS is a lightweight procedure. This instructional exercise covers the vast majority of the points required for an essential comprehension of PhantomJS. Moreover, this instructional exercise likewise discloses how to manage its different segments and to get a vibe of how it functions. We are using jasmine and karma plugin of gulp task runner. And we make the test-file for to design all the test in test-suits. By the help of karma tool, we validate the source code against those tests which are in test-suits. Also maintain the karma.conf.js file to set the configurations like basePath, frameworks, files, exclude, reporters, port, colors, logLevel, autoWatch, browsers, captureTimeout, singleRun and task execution fires from gulpfile.js and its result will be shown on browser. And another browser window will also automatically open. There is one button called debug, when we press that button then we can show output of that GUI code. [15, 16]

This is a generic for all type of application like web application, desktop application, and mobile application and domain of these application are Android, iOS, AngularJs, and those applications are can on any digital devices and on IoT devices like PLCs, Set-Top boxes, Television set, mobile, router, modem etc. Which have already assigned IP address. And these applications are input for this gulp task runner file. Using open source plugins, and self-made gulp task runner plugins we are implementing functionality as per our needs. Many of plugins for some functionality are readily available and some plugins, we must build as our project specific requirements. Here, unit testing part is start before the development part. Because of, at the time of requirement specification, our automated unit testing part is starting. One time needs of target application is fixe, and then we can write tests in test-suites. This is a part of extreme programming. In our use case, all the requirements are present in the form of test in the code. By the help of Karma tool, Jasmine frame work, and phantomJS we can start our unit testing scenario. By the help of this we can starts background operation related to unit testing and we can say that in our case, unit testing is start before starting of the development phase of application. Here, we can say that development procedure of application is test-driven. Development procedure goes as per the guidance of result generated by tests from the test-suites. And we are using TDD procedure for development of applications. TDD is test-driven development; this term is coming from the area of extreme programming.

This unit testing phase is continuously monitoring or watching updating development source files of application from application source module. If any changes or modification is detected then it is directly reflected on the browser window and result is also we can see on command prompt with reason of result of tests. There is one new browser is also automatically start which gives the information about that Karma tool is already started. And by the help of debug button we can check our application, which is developing. So, cycle of TDD is working in this part of project. So, by this approach our application development phase includes whole unit testing phase. Here, we can save time which invest between development and testing phase.

In figure 3.1, we are showing example Gulpfile.js. It represents the example structure of gulp file. Here, 'gulp' and 'gulp-karma' both are plug in of Gulp task runner. And, we are directly using this plug-in functionality in this Gulpfile.js file, which is written in JavaScript format. Here, 'gulp.task()' is function of Gulp task runner, which is use for to design the task body and map with some unique name over Gulpfile.js. So, this task body is mpped with name 'default'. 'gulp.src()' is also function of Gulp task runner, which is use for to give reference of source code file package of application, which we wants to process by gulp file with task name 'default'. '.pipe()' is use for to create a pipe for paralleling execution of tasks. In this, we are using 'Karma' as a tool with two parameters and their values like 'configFile : 'karma.conf.js". And, 'action : 'watch" represents that it any changes will make in source file of application, then it will directly detect those changes and according to those changes it gives response to process.

In figure 3.2, we are showing one small browser window with browser name 'Chrome 51.0.2704' and it represent the state of browser is idle. It also gives the information about that this browser is automatically open by 'Karma v0.13.22 tool', which is the result of execution of 'Karma.conf.js' by executing the task 'default' in file 'gulpfile.js'. It is open in 'Chrome' browser because declaring the 'browser : 'Chrome" in 'Karma.conf.js' file. And, that we can see in the 'karma.conf.js' file. We can use other browser like 'PhantomJS' and others otherwise it will catch default browser. That 'DEBUG' button is also very useful for us. Because, by the help of this we can debug our application, whose source code package is processing here. And, it will run till execution of 'default' task.

In figure 3.3, 'karma.conf.js' is a main configuration file to do this unit testing module. And it has many configuration setup parameters like basPath, files, autoWatch, frameworks, browsers, plugins, junitReporter, etc. It shows parameters and their appropriate values. this is a key configuration file, which work as a bridge between source code files packages and test JavaScript files, which use to testing. And, by this file execution we can plotting the result of test cases on the given browsers. This basePath parameter specifies source code files packages path. And, files parameter specifies files packages path of related to test-suits and other libraries. If this autoWatch parameter has value true, then it will work same as gulp task runner function 'gulp.watch()'. This part of project also needs tool like Karma as well as framework like jasmine, chai, mocha, etc. Here, we are using framework 'jasmine'. Browser parameter gives the value about the browser we are using for plotting results of test-cases by JavaScript test-suits. plugins parameter takes the values in a form of name of plugins, which we are using in unit testing phase. And junitReporter gives reference about the task output file of result of test-cases.

In figure 3.4, we case the result of test-cases from test-suit JavaScript files. On the top we can see the name of framework 'jasmine 1.3.1'. Besides it we can the revision number it is unique for each attempt for testing. Besides it shows the processing time of testing by 'finished in 0.12s'. And after that we can see the dots. This number of dots gives the information about the total number of test-cases. And gives the information about the how many test-cases are pass and how many test-cases are fail. If dot color is green means that test-case is pass, and when dot replace with the red cross then it gives the information about the test is fail. After this it gives how many numbers of test-cases are pass by 'Passing 12 specs'. And it has facility of try/catch if any test-case is fail. After those statements gives the test-suit names and test-cases names.

Goal: We are developing the support code or raw code structure for any application. So, we develop code structure in such a way, hence we can generically use in different-different application development procedure. In this current phase we are focusing on generic code development for development and unit testing.

Gulpfile

Gulpfile.js

```
// gulpfile.js
var gulp = require('gulp'),
    karma = require('gulp-karma');

gulp.task('default', function() {
  gulp.src(['src/js/**/*.js'])
    .pipe(karma({
      configFile: 'karma.conf.js',
      action: 'watch'
    }));
});
```

Figure 3.1: Gulpfile[2]

Debug Browser Window

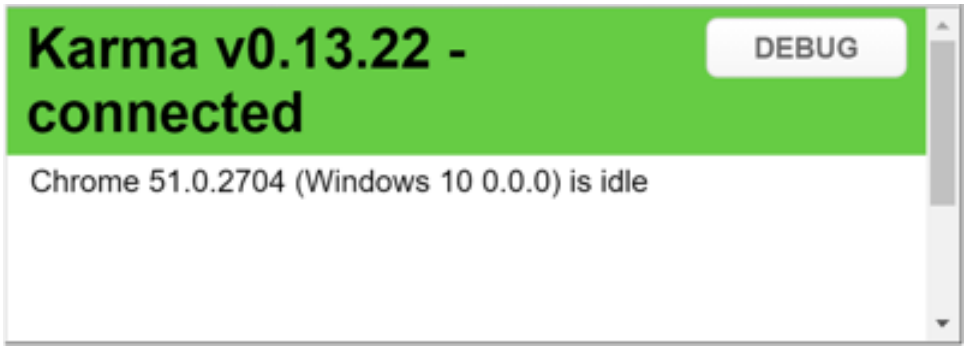

Figure 3.2: Debug Browser Window[3]

’karma.conf.js’ File

karma.conf.js test

```
module.exports = function(config){
  config.set({

    basePath : '../app',

    files : [
      'lib/angular/angular.js',
      'lib/angular/angular-*.js',
      '../test/lib/angular-mocks.js',
      '../test/lib/sinon-1.15.0.js',
      'js/**/*.js',
      '../test/unit/**/*.js'
    ],

    autoWatch : true,

    frameworks: ['jasmine'],

    browsers : ['Chrome'],

    plugins : [
      'karma-chrome-launcher',
      'karma-jasmine'
    ],

    junitReporter : {
      outputFile: 'test_out/unit.xml',
      suite: 'unit'
    }

  });
};
```

Figure 3.3: ’karma.conf.js’ File[4]

Result Of Tests

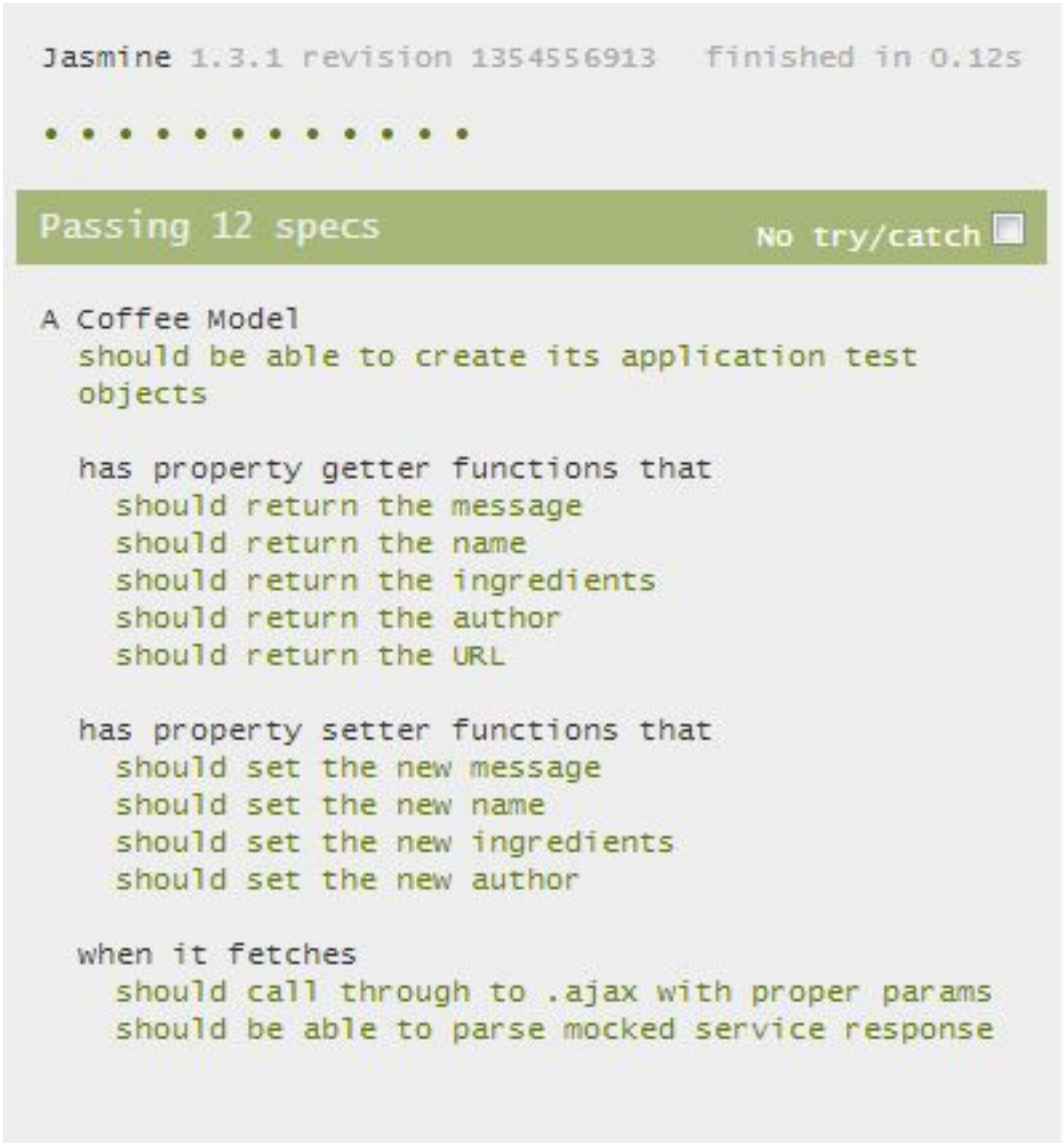

Figure 3.4: Result Of Tests[5]

3.2 Production & code release:

Production code release phase takes raw code structure for to increase efficiency in both directions like the size of the project or individual project module and execution time of the application also focusing on that how code compilation take less amount of time and reduction in unnecessary number of compilation task which is included in the project execution life cycle. Benefit of this part of project is to improve optimization in the case of source code of applications. So, by use of some task runners we can reduce both size and running or execution time of the project. Using gulp task runners with the use of built in plugins, and new plugin we can solve the problem of development and production in software or application life cycle like minifying, uglifying, concatenation of files, rename, versioning, task-re-usability, source-mapping, image optimization, encryption, pipe re-usability, compression, vulcanize and helper for minimizing the effect of error, faster cache accessing. [17, 18, 19]

Here, we can make code as efficient as possible. So, it will take less memory space requirements. And, also increase the speed of source code of application. Here, we can also apply some functionality. So, source code will be in non-readable format. And, source mapping functionality we can apply to source code of application. Compression functionality we can apply on our source code of application. By this we can save our memory rooms for other purposes. If we want to make good light weight source code of application then this approach is very important. When we apply this phase to source code of application response time can be decrease of application.

Here, in the figure 3.5, we can see the example of Minifying HTML file by Gulpfile.js using of Gulp task runner. And 'gulp-changed' and 'gulp-minify-html' plugins are using with variable names like 'changed' and 'minifyHTML'. Using this variable name, we can the functionality of those plugins. Then, we are creating the task with name 'htmlpage', after this 'gulp.src()' takes the source application folder. 'minifyHTML()' actually minify the all referenced HTML files. And 'gulp.dest()' makes new source code application package and put it in to destination location.

This is just an example, as per understanding from this example we can implement other Gulp functionality by its respective Gulp plugins. Uglify, rename, imagemin, plumbering, helper, watching, concate, vulcanize, revisioning are feature of Gulp task runner that we can implement by plugins with this method. By the use of this Gulp task runner features, we can achieve faster and optimized source code of application with very less errors.

Goal: We are minifying and optimize available code and just make it efficient. So, we can increase the speedup of a target application in which we are using this phase. We have generated the production code in such a way, hence we can generically use in different-different application development procedure.

Minification Of HTML File

Figure 3.5: Minification Of HTML File[6]

3.3 Data creation:

Data creation involves both data storage data execution. In this, many activities are included like to identify useful and relevant data attribute from the huge amount of data sets or data warehouse and data retrieval, insertion/update operation on data. Storage place is a cloud. So, cloud queries are also in this phase. And cloud data is a super set of all storage of data in our project. And when we are in data analytics phase that many device data are store on local controller means FOG. Those devices are IoT devices like PLCs, routers, modem, phone, set-top boxes, Television, etc. These types of data also describe the health of devices, which are connected over the network. We are using JSON and text format and MongoDB and SQL structure for to deal with the data.[17, 20]

Data creation phase also involve how to collect data from where. So, data creation has job for to investigate for, from which virtual or physical location over the network or at the boundary point of network our target data or information is available. And which types of network devices are important for us in this data availability. Also, the point of analysis is that out of the total data set, which network device plays how much contribution to total data set. In this analysis, time constraint or time interval or time duration is also important, when important data or information is at important network devices. Also storing and retrieval strategy to/from that network from/to Iot devices. Also, it is important that where we want to store collected data, and which type of functionality we want from storage area like structured or unstructured database. Duration of time for storing and retrieval of data or information to/from storage area is also very important factor. And which data or information we want to store on cloud and which type of data or information we want to store on local controller of organization.

These are type of protocols, which we can use for this task. one is SNMP and second is TR-069. Simple Network Management Protocol and Technical Report - 069 are used for management of customer premise equipment in network environment. Basic method for to manage the customer premise equipment in network environment is remote procedure call by organization to customer premises. There is a set of data models of both protocols. And those data model describes procedure/method, which is in the form of parameters or attributes. All customer premise equipment in network environment or embedded devices are maintaining their configuration file for those attribute or parameter's values. So, we can say that technical person can remotely get and set the parameters/attributes in configuration file of customer premise equipment. In this scenario, we are also take care about the security of those customer premise equipment. If we do not take care about this scenario, then it makes security violence cases. And all customer premise equipment can control by any third party. So, we are using one middle entity called auto configuration server. This auto configuration server is responsible for authentication of technical person as well as customer premise equipment. This auto configuration server authenticates technical person by proper log in identification and password from organization domain. And organization maps customer premise equipment or embedded devices by authentication credentials. After this technical person can communicate with customer premise equipment or embedded devices by remote procedure call method. And this is a type of CPE WAN Management Protocol (CWMP).

CPE WAN Management Protocol

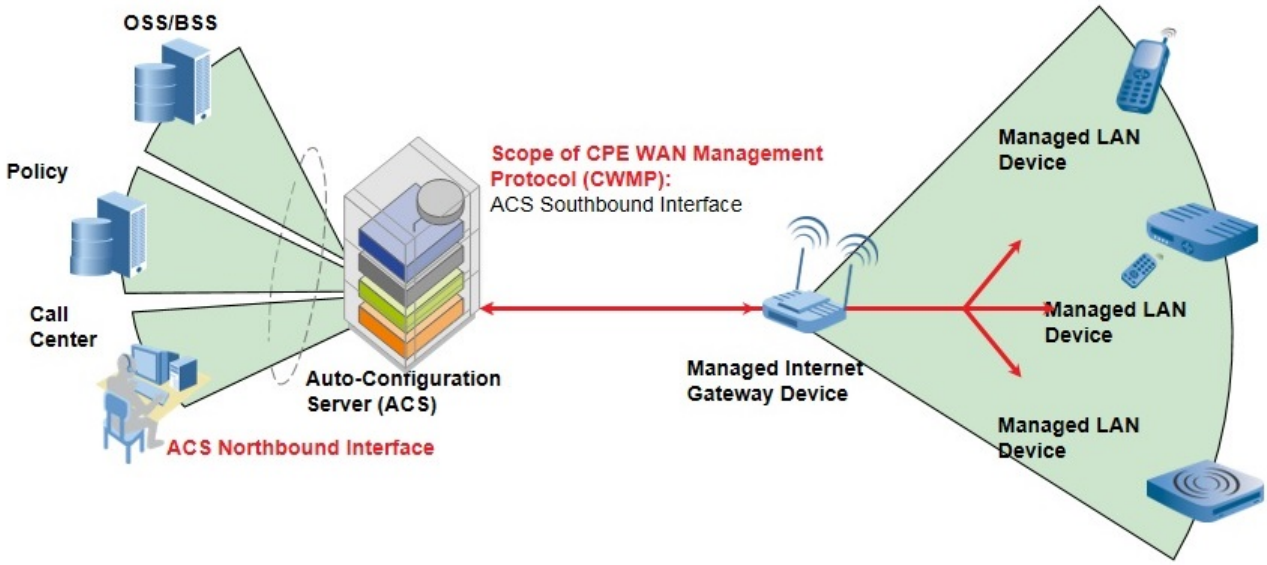

Figure 3.6: CPE WAN Management Protocol[7]

Here, we can see both are network management protocols. But, they have capabilities of to get and set the value of parameter of attributes in configuration file of customer premise equipment or embedded devices. So, we are using these types of protocols for to get or set the information/data/value of customer premise equipment or embedded devices. But, we are using the TR-069 protocol for to implement this part of project. It is an application layer protocol.

Because, there are many advantages of TR-069 over the SNMP protocol. It has functionality of high security and scalability. It gives high degree of security of customer premise equipment/embedded devices as well as auto configuration server. It has also large list of good functionalities and more numbers of parameters or attributes. It gives dynamic services from a central entity. TR-069 has many data model for different purposes. like TR-181, TR-104, TR-098, TR-106, TR-143. TR-181 is use for device data model and figure 3.8 describing it. TR-104 is use for VoIP. TR-098 is use for Gateway device data model. TR-106 is use for Baseline data model. TR-143 is use for Enabling Network Throughput Performance Tests and Statistical Monitoring data model.

As we can see in figure 3.6, TR-181, TR-143, TR-098 are more important for our next data analytics part. Because, it provides the profile of customer premise equipment or embedded devices with TR-181 for to get or set the value of parameters/attributes of configuration file of embedded device , profile of Gateway device with TR-098 for to get or set the value of parameters/attributes of gateway devices, and profile of network throughput performance and statistical monitoring with TR-143 for monitoring performance and throughput.

Now, we are focusing on architecture of forth part of project. But, how can we use this third part in forth part that we can see in figure 3.7. Here, we can see gateway (FOG Model) and embedded devices are communicating with each other using Remote Procedure call by TR-069 protocol. Then, gateway communicates with cloud (Cloud Model). First embedded devices are control by its own configuration file. So, it is called control priority rank 1. secondly, local controller or gateway or FOG model control embedded devices by modification in its configuration file. So, it is called control priority rank 2. And third, local controller or gateway or FOG model fetch or use the configuration file from cloud and then put into or replace with configuration file of that embedded devices. So, it is called control priority rank 3. we can see the control priority decreasing from bottom to top and data storage capacity is increasing from bottom to top.

Goal: We are focusing on type of data, retrieval, and insertion/update operation on data and selection of data and determining the usefulness of data. Also, identifying important data or information source network devices and where they are placed logically and physically.

Data Models

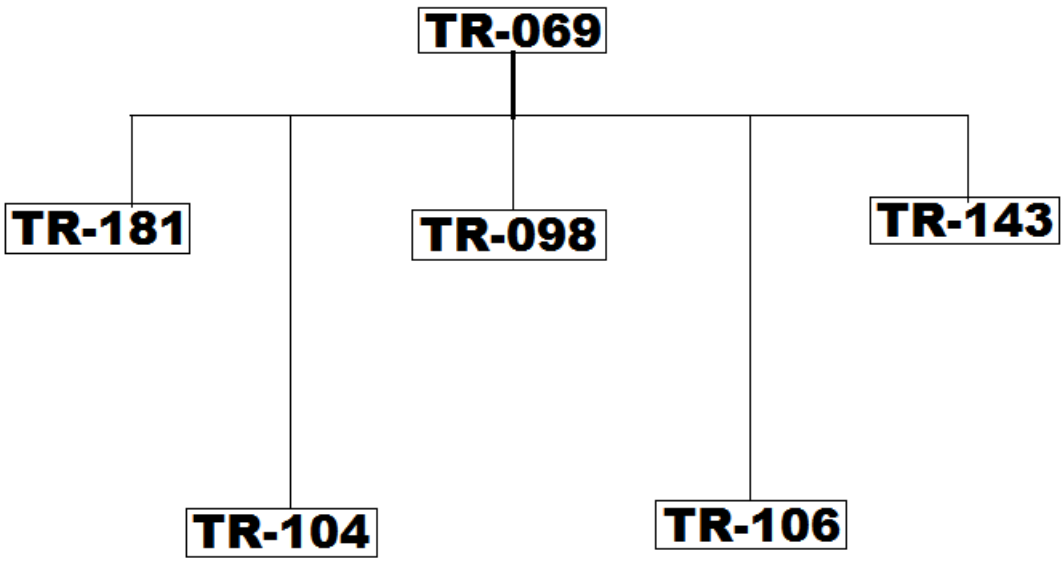

Figure 3.7: Data Models

Architecture

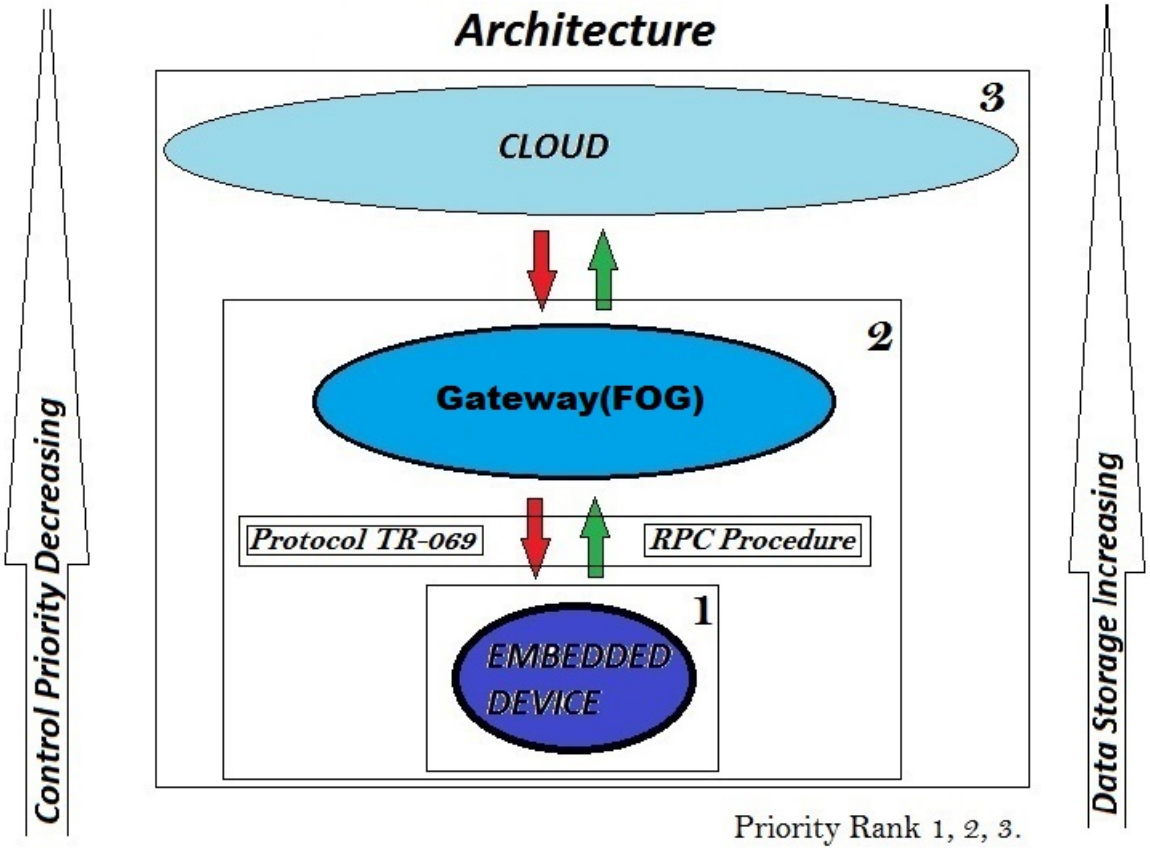

Figure 3.8: Architecture

TR-181 Device Data Model Profile

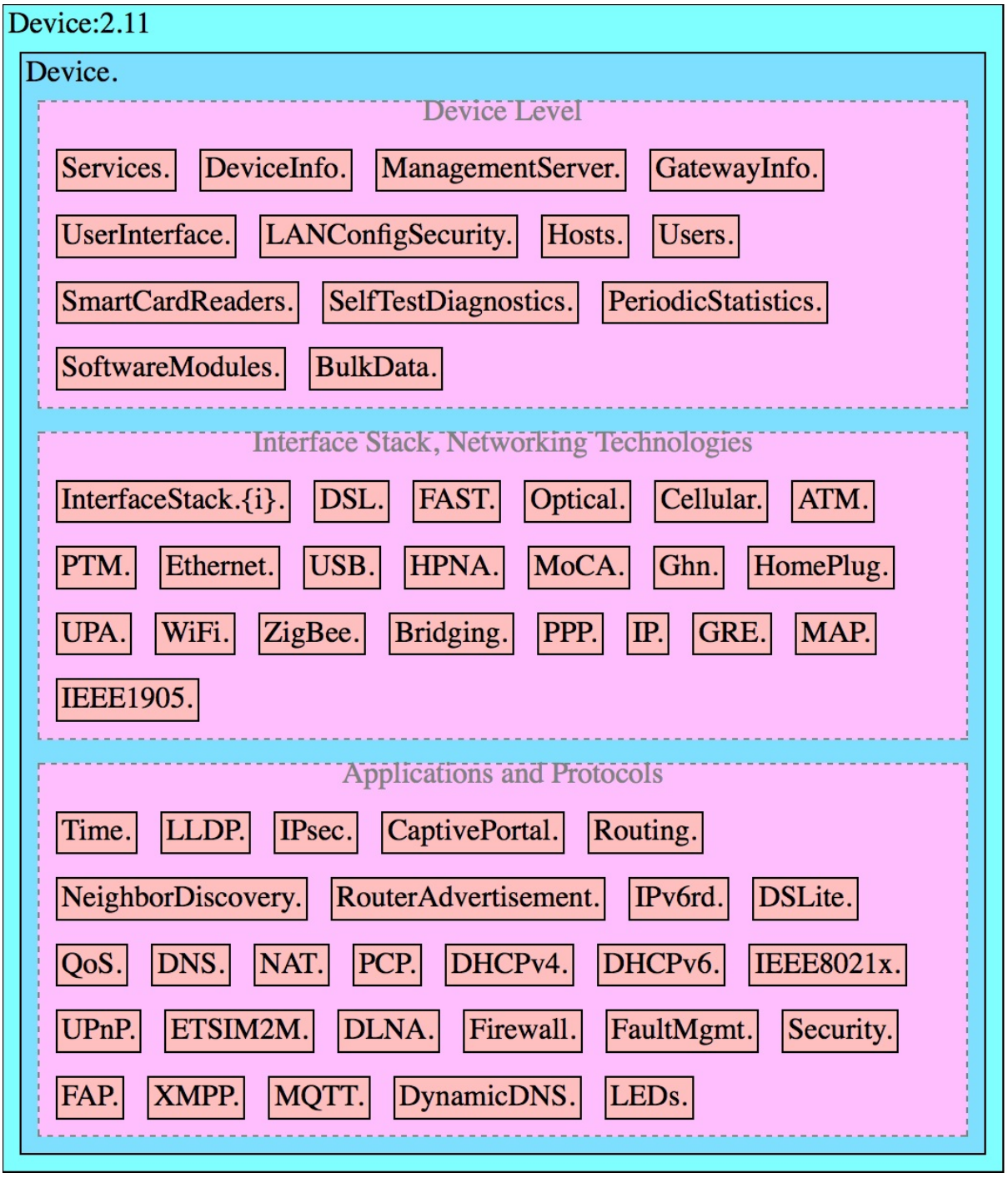

Figure 3.9: TR-181 Device Data Model Profile[8]

Data Creation Implementation Example

```
import com.cwmp.acsserver.acs.structrpc.InformResponse;
import com.cwmp.acsserver.acs.structrpc.SetParameterValues;
import com.cwmp.acsserver.acs.structrpc.SetParameterValuesResponse;

public class CpeClient implements ICpe
{
        private String username;
    private String password;
    private String ipAddress;
    private int port;
    private String alias = "";
    private String connectionUrl = "";
    private CpeInformation cpeInformation;
    private Tr69Communication comm = null;
    /**
     * initialized in constructor to 10
     * can be re-initialized by setter
     */
    ArrayBlockingQueue<AcsConfigurer> arrayBlockingQueueForRpcTasks;

    public CpeClient() {}
    public CpeClient(CpeInformation cpeInformation)
    {
        this.cpeInformation = cpeInformation;
        this.comm = new Tr69Communication(this);
        this.arrayBlockingQueueForRpcTasks = new ArrayBlockingQueue<AcsConfigurer>(10);//10 RPC buffer allowed, can be re-initialized by se
    }
    public CpeClient( CpeInformation cpeInformation, String ip, int port )
    {
        this(cpeInformation);
        this.ipAddress = ip;
        this.port = port;
```

Figure 3.10: Data Creation Implementation Example[9]

3.4 Data analytics:

We are using gulp task runner for to insertion/updation and retrieval of data to and from devices of network. We have big amount of historical data about the data transmission of data over the network. We have data analytics techniques for to clean up the row data and to smooth the historical data which is store in data warehouse. Then, we execute the very important part of data analytics, which the selection of attributes which are directly or indirectly plays role for value of final network throughput. This phase is totally dependent on previous data creation stage. And accuracy of prediction or forecasting and performance of whole data anaytics are also totally dependent on previous data creation stage . If we select appropriate database and its storing and retrieval policy then this stage can perform good. This stage also dependent quality of data, which we are using as training set. And selection of good algorithm in case of response time, memory space and accuracy of forecasting.

And after that we are focusing on problem of traffic, packet loss, etc. these all problems are trying to solve by data analytics and all its activities are automated through the task runners. Like all data retrieval, data storing and/or data update during data analysis. Data analysis may be done either locally in centrally located devices (FOG) or remotely in servers (CLOUD). With automated task runners used in data analysis we can then focus our effort more to understand the behavior of data traffic, failure analysis and according adapt with appropriate action such as improving the throughput of the whole system.

This all activities are doing automated by the help of gulp task runner. So, all tasks are doing their job in background automatically. According to the result of data analytics algorithm we set the automated system for to apply the solution over the network by using gulp task runner. By using this solution we can improve the throughput of network.[20].

Use-case : WiFi Band Steering

This is a use-case of Data analytics. Which is related to WiFi systems. And entities, which are involve in this 'WiFi Band Steering' are dual-band WiFi deployments and dual-band client IoT devices like PLCs, routers, modem, phone, set-top boxes, Television, etc. It will push the dual-band client IoT devices to connect less-congested and higher capacity 5 GHz band from more-congested and less capacity 2.4 GHz band and stop the dual-band client IoT devices attempts to connect with the 2.4 GHz band in the situation of congestion means more dual-band client IoT devices are trying to connect less capacity 2.4 GHz band, when higher capacity 5 GHz band not completely utilized.

It is compulsory for both WiFi deployments and client IoT devices or embedded devices are in dual-band as 2.4 GHz band and 5 GHz band.

Reduce Wi-Fi Congestion With Band Steering

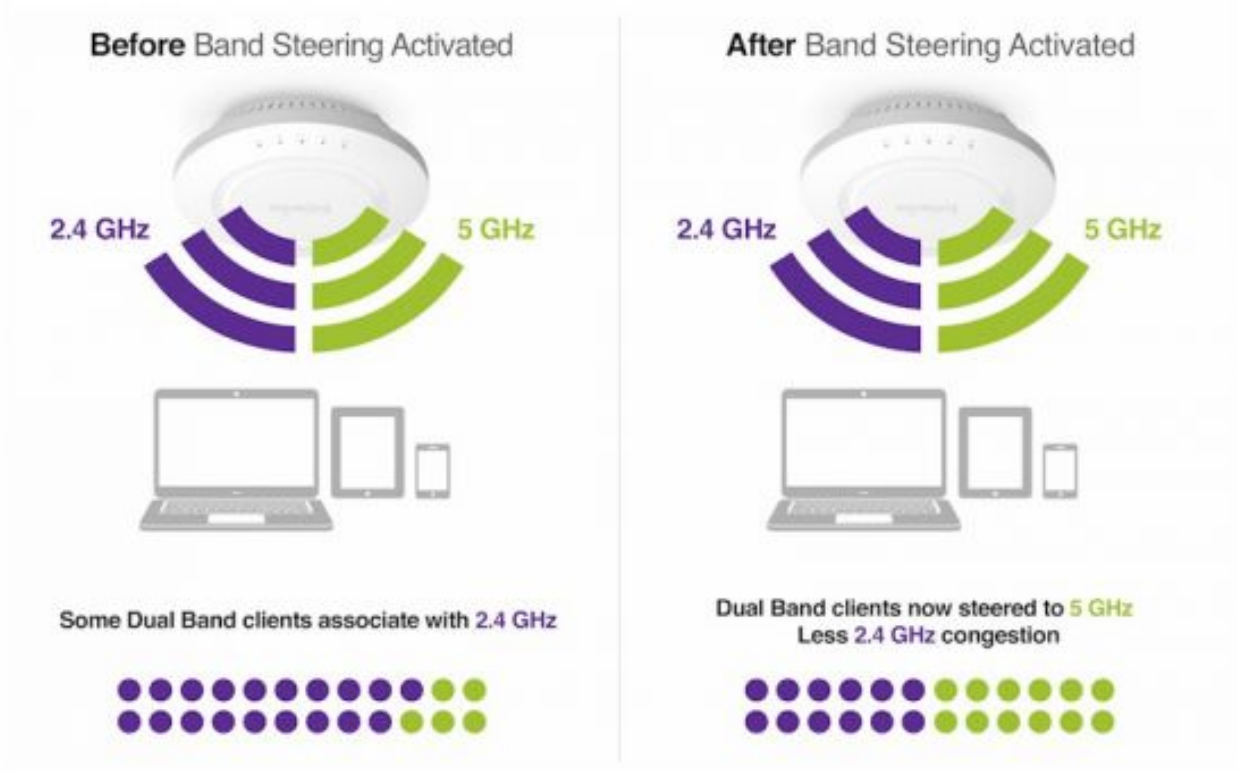

Figure 3.11: Reduce Wi-Fi Congestion With Band Steering[10]

Frequency Bands:

2.4 GHz :

2.4 GHz frequency band is used by the older wireless standards (802.11 b/g/n). 2.4 GHz frequency band has furthest distance capacity, but speed is very low. This 2.4 GHz frequency band is also the most vulnerable to interference from other wireless network or WiFi access points. We can say that 2.4 GHz frequency band has two types of difficulty. first, 2.4 GHz frequency band has narrow band and less speed. Second, 2.4 GHz frequency band has interference problem with other wireless network or WiFi access points. So, we can say that when our priority is long distance, then 2.4 GHz frequency band will win over the 5 GHz frequency band with few number of dual-band client IoT devices like PLCs, routers, modem, phone, set-top boxes, Television, etc for to connect in long distance.

5 GHz :

5 GHz frequency band is using the wireless standards such as 802.11ac and old standard 802.11 a/n. 5 GHz frequency band has nearest distance capacity, but speed is very high. This 5 GHz frequency band is not vulnerable to interference from other wireless network or WiFi access points. We can say that 5 GHz frequency band has two types of benefits. first, 5 GHz frequency band has broad band and high speed. Second, 5 GHz frequency band has not interference problem with other wireless network or WiFi access points. So, we can say that when our priority is high speed, then 5 GHz frequency band will win over the 2.4 GHz frequency band with more number of dual-band client IoT devices like PLCs, routers, modem, phone, set-top boxes, Television, etc for to connect in long distance.

Comparison Of 2.4 GHz vs. 5 GHz Frequency Bands

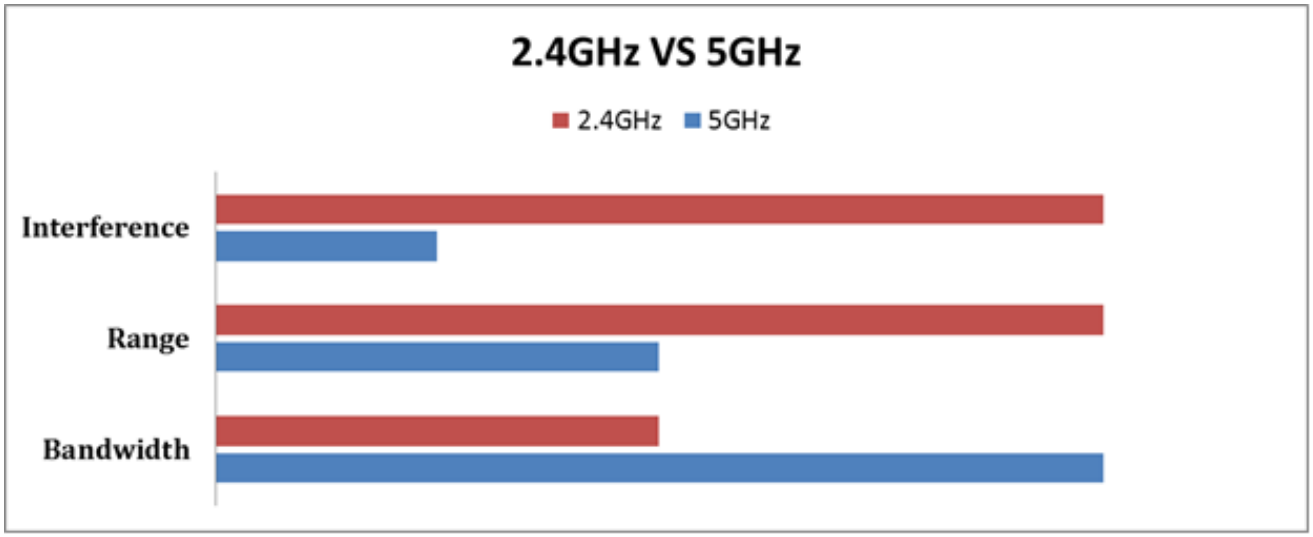

Figure 3.12: 2.4GHz vs. 5GHz[11]

WiFi Band Steering Scenario

For understanding, we are taking one scenario of WiFi Band Steering. We can see here dual-radio access point, which is broadcasting 2.4 GHz band and 5 GHz band. And there is two SSID. One is 'Internal' and second is 'PBSPOT'. The SSID 'Internal' providing both 2.4 GHz and 5 GHz frequency bands. But, second SSID 'PBSPOT' providing only 2.4 GHz frequency band.

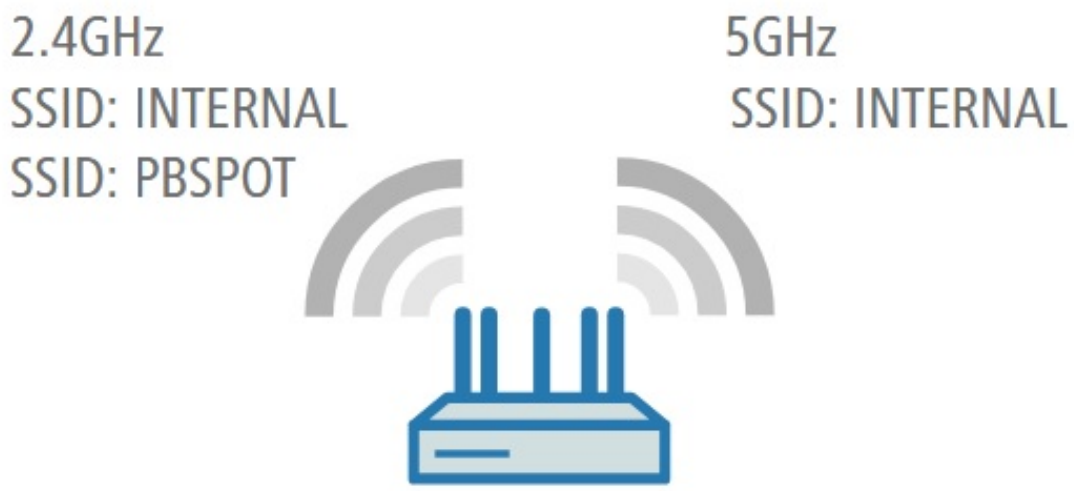

Figure 3.13: Dual-radio access point[12]

Status Before WiFi Band Steering Of Client Devices:

We can see here, there are two types of frequency bands are available for to connect the WiFi 2.4 GHz frequency band and 5 GHz frequency band. We also have two SSID. One is 'Internal' and 'PBSPOT'. Total number of client devices are 20. Three client devices are in 'Internal' SSID with 5 GHz frequency band. And the 2.4 GHz frequency band two SSID are exist, one is 'PBSPOT' SSID with five client devices and second is that 'Internal' SSID with twelve client devices.

Frequency	2.4 GHz		5 GHz
SSID	PBSPOT	INTERNAL	INTERNAL
Clients	5	12	3

Figure 3.14: Before WiFi Band Steering[12]

Status After WiFi Band Steering Of Client Devices:

We can see here, after WiFi band steering of client devices, eight client devices from SSID 'Internal' and 2.4 GHz frequency band steer to SSID 'Internal' and 2.4 GHz frequency band.

Frequency	2.4 GHz		5 GHz
SSID	PBSPOT	INTERNAL	INTERNAL
Clients	5	4	11

Figure 3.15: After WiFi Band Steering[12]

Problem In WiFi Band Steering Of Client Devices:

When we apply both 2.4 GHz WiFi frequency band and 5 GHz WiFi band with same SSID. And then, we connect the client device. It automatically use the logic that when client device near to WiFi source then it will connect with 5 GHz WiFi frequency band and it will connect with 2.4 GHz WiFi frequency band, when client device far to WiFi source, without caring about congestion of client devices in WiFi frequency band.

Solution Of Above Problem In WiFi Band Steering Of Client Devices:

This problem is arising because of 2.4 GHz WiFi frequency band and 5 GHz WiFi frequency band both belongs to same SSID. So, solution is to design separate SSID for two different frequency band 2.4 GHz WiFi frequency band and 5 GHz WiFi frequency band.

Procedure Of WiFi Band Steering Of Client Devices:

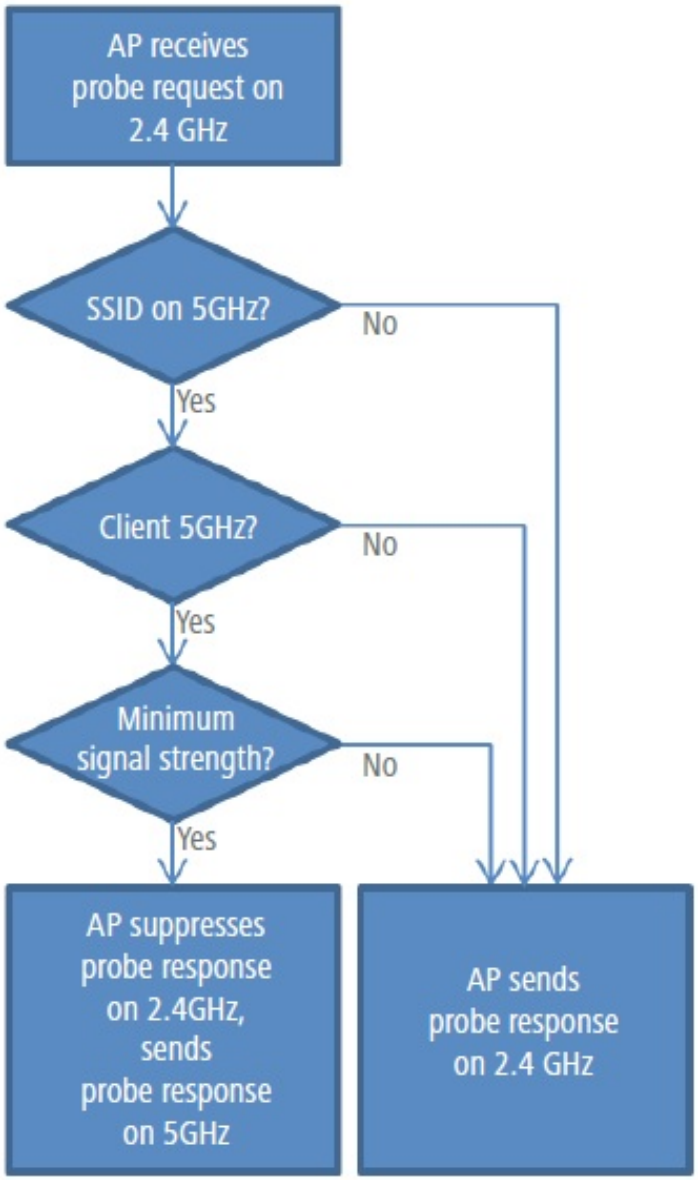

Figure 3.16: Procedure Of WiFi Band Steering Of Client Devices[12]

Initially all WiFi Band requesting for to connect with 2.4 GHz WiFi frequency band. This is first step of the procedure Of WiFi Band Steering of Client Devices. Then, we check has 5 GHz WiFi frequency band or not. If it has 5 GHz WiFi frequency band capacity then after we checks client devices has capacity of 5 GHz WiFi frequency band or not.

If client also has capacity of 5 GHz WiFi frequency band then after we checks signal strength between SSID and client device are minimum or not. If all these conditions are true or 'yes' then we switch client device from 2.4 GHz WiFi frequency band to 5 GHz WiFi frequency band for best utilization of bandwidth. If any one of those condition is false or 'no' then we establish connection through 2.4 GHz WiFi frequency band between SSID and client devices. This is basic procedure that how WiFi Band Steering is working.

Data Analytics Implementation Example

```
import static org.junit.Assert.assertEquals;

import org.jdmp.core.algorithm.classification.Classifier;
import org.jdmp.core.algorithm.classification.bayes.NaiveBayesClassifier;
import org.jdmp.core.dataset.CrossValidation;
import org.jdmp.core.dataset.ListDataSet;
import org.junit.Test;
import org.ujmp.core.listmatrix.ListMatrix;

public class TestNaiveBayesClassifier {

        @Test
        public void testIrisClassification() throws Exception {
                ListDataSet iris = ListDataSet.Factory.IRIS();
                Classifier c = new NaiveBayesClassifier();
                ListMatrix<Double> results = CrossValidation.run(c, iris, 10, 10, 0);
                assertEquals(0.959, results.getMeanValue(), 0.04);
        }

}
```

Figure 3.17: Data Analytics Implementation Example[13]

3.4.1 Data exchange with presence of cloud:

It involves three major entities home network controller, cloud storage, set of home devices hierarchy. These all three entities exchange the data as per their needs and fulfillment of requirements. Data set relation of these three entities: cloud storage greater than home network controller greater than set of home devices hierarchy.[17]

There is cloud model implementing the data analytics phase. We are using organizational cloud. At that cloud point can directly connect with local controller server. And, the cloud server and server of local controller can connect through the internet or organizational intranet. So, cloud server can access the data of server of local controller as same as data accessing locally from local controller. So, we can say the now data transfer is faster and there is no dirty read problem with data means data is consistent. Because, FOG computational model and Cloud computational model both transferring data to/from same source. So, here two problems are solved simultaneously. Cloud server can directly interact with server of local controller with related credentials. So, we are also saving the memory space and data transfer time. It generates the prediction in the form of Will client device steer or not? as per analysis of historical data. And take proper action as that conclusion.

Goal: As per needs, data exchange activities are initiated within these three entities. Because needs generate by data mining. Data mining uses these data and describes the behavior of the system and forecast information which is useful to improve the throughput. Here data mining doing at the cloud levels.

Architecture With Cloud Model

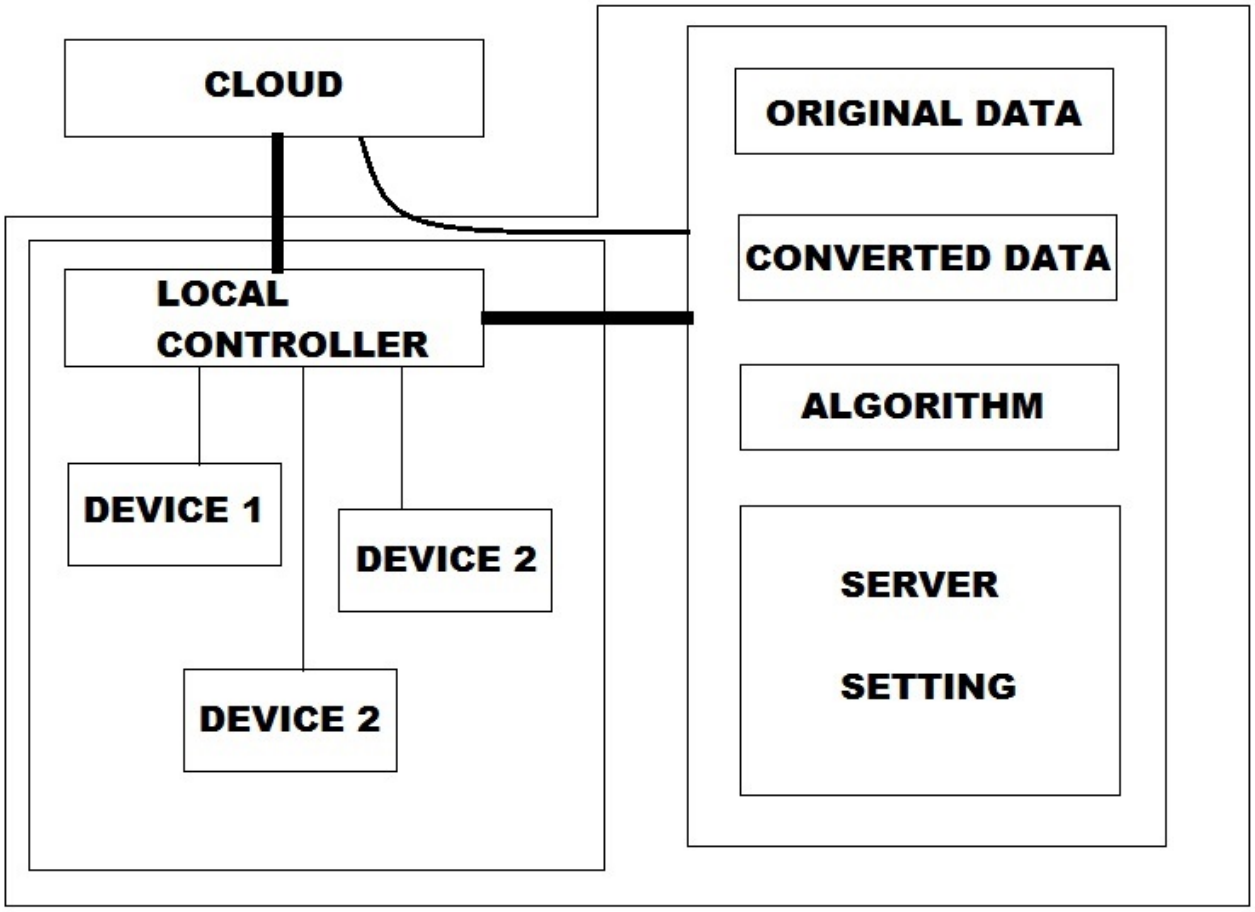

Figure 3.18: Architecture With Cloud Model

3.4.2 Data exchange with absence of cloud:

It involves two entities home network controller, set of home devices hierarchy. These two entities exchange the data as per them needs and fulfillment of requirements. Data set relation of these two entities home network controller greater than set of home devices hierarchy. Here, we are using concept of FOG. The objective of fogging is to enhance productivity and diminish the measure of information transported to the cloud for handling, examination and capacity. This is regularly done to enhance effectiveness, however it might likewise be utilized for security and consistence reasons. Since cloud computing is not practical for some internet-of-things applications, mist computing is often utilized. Its appropriated approach tends to the requirements of IoT and modern IoT, and additionally the massive measure of information shrewd sensors and IoT gadgets create, which would be expensive and tedious to send to the cloud for preparing and investigation. Mist computing diminishes the data transfer capacity required and decreases the forward and backward correspondence amongst sensors and the cloud, which can contrarily influence IoT execution. It generates the prediction in the form of Will client device steer or not? as per analysis of historical data. And take proper action as that conclusion.[17]

Here, we are implementing the solution of WiFi band steering of client devices by the help of data analysis. We are using those data set which are collecting from the devices by the help of TR-069 protocol profile. All those data are currently in the server of local controller. Here, we are filtering the data or converting the original data in to some format therefore we can apply predictive analysis models. These converted forms of data also store in that server of local controller. So, server of local controller has two profiles of data first is original profile of data and second is filtered or converted form of data for to do analysis and both data profile is stored in same location. Here, all the data and algorithm both are stored in the server of local controller. This algorithm is predictive analysis algorithm. Local controller processes the historical data by this predictive analysis algorithm and generate the solution for WiFi congestion in the form of WiFi band steering. Here, all those data are stored in structured format. So, we can use structured query language (SQL). As we can see the whole picture in the figure 3.16.

We are implementing this whole scenario with help of java programming language. Because, we are very familiar with the java programming languages. Second, java is platform independent and world wild using. And java programming language has good number of packages for to implement this scenario. So, there is many packages for to implement data structure related to data analytics. Those packages are available also for to implement the functionality of those analysis algorithm. So, here we can directly train the model with help of historical data because the algorithms are already inbuilt and data structure are already inbuilt in java packages related to data analysis.

There are many libraries are available for data structure, data analytics algorithm and data operations. So, we can directly use those library or packages for to implement last phase of this project. Weka is java based data analytics tool. Same as Weka tool we can also implement data analytics phase in WiFi band steering use case.

Goal: As per needs data exchange activities are initiated within these two entities. Because needs generate by data mining. Data mining uses these data and describes the behavior of the system and forecast information which is useful to improve the throughput. Here, data mining doing at the local level means at home network controller.

Architecture With FOG Model

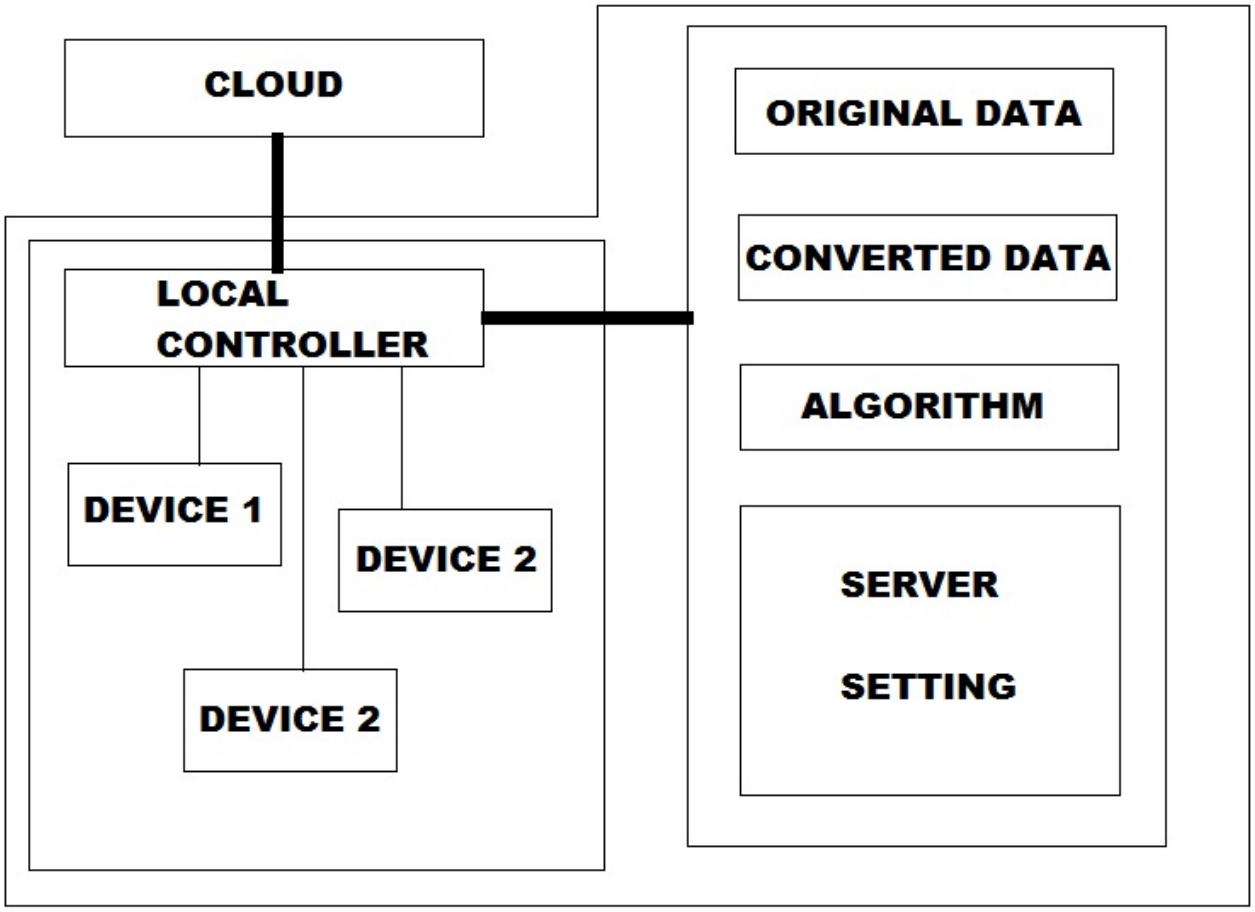

Figure 3.19: Architecture With FOG Model

Chapter 4

Technology

4.1 Gulp:

Gulp task runner uses five basic functions like task () which is use for to define the body of task, src () which is use for to referencing location of source files, dest () which is use for to referencing destination location of where we want to build target files, watch () which is use for to observing or to monitoring the file or file collections about the modification or changes, start () which is use for to start the execution of specific task out of set of tasks. If some task we want to do repeatedly, with less effort and less amount of code. If we are developing the new project or software project, then many of those activities we must do repeatedly like optimizing images, minifying code, and compiling files. We can use these all tasks, and task runners contains the behavior of tasks in a form of code in the task runners file like gulpfile.js if we use gulp task runner. And according to the developer logic, Developer set the sequence of execution of those tasks which is handles by the task runners. [17, 18]

1. Minify: This task is use for to minification of source code of files like html and css. All the spaces are removed from source files.

2. Uglify: This task is use for to uglification of source code of files like JavaScript. All the source code is converted in to not readable form.

3. Rename: This task is use for to rename the source file. When we concatenation of two or more than two files by the use of this.

4. Imagemin: This task is use for to optimizing size of image source files like png, jpeg, etc. and we can save memory.

5. Plumbering: Plumber is use for to masking the errors effect. When any error is occurring, then gulp execution pipes may be break. But, by using of Plumbering we remove those effects.

6. Works as a helper: An official set of helper functions to use for to color console output.

7. Browser Synchronization: This task is for to Synchronized browsing across multiple devices, and refreash automatically when changes are detecting in source files.

8. Watching: this task is for to monitor or to detect any changes are occur in source file of application.

9. Concate: This task is use for to concate css files. It is also use for to concate JavaScript files.

10. Vulcanize: This task is use for to valcanize pipes of gulp execution, which is generally break by occurrence of error.

11. TDD: This task is use for to apply unit testing on application module.

12. Revisioning: This task is used for to Revision of all the files in the set of application source files.

4.2 TDD:

We are applying here TDD methodology means Test driven development for optimizing time in testing phase of application life cycle. Test-driven development is application development procedure which is applying on module of application or software in small size. All the needs or requirements are representing in test cases of test suits. Test driven development is related to the test-first programming, which is concepts of extreme programming. So, all the needs or requirements are verifying at the time of development of that application module. So, this process saves the time which is vesting in module check at testing phase. Here, developer plays role more than developer. By using of Test-driven development, developer understand or realize his or her mistakes very soon. So, we can optimize the time for application development in application life cycle.

4.2.1 Generate a new tests which defines requirements of development

In a test-driven development, all the features or needs or requirements are in the form of test cases, which is managing in test suit files. Each test in test suit defines the function, and in that function needs or requirements are in the form of code and developed code are check against these tests which is define in test suits. To write a test, the developer must clearly understand the features specification and requirements. The Test-driven development makes the developer focus on the requirements before writing the code.

4.2.2 Check all the tests for examination in test suits

This affirmation or supports that the experiment in test suits is working precisely, exhibits that the new test does not abandon requiring new code in light of the fact that the required direct starting at now exists, and it chooses out the probability that the new test is blemished and will constantly pass. The new test should fail for the ordinary reason. This movement fabricates the software engineer's trust in the new test.

4.2.3 Write correct or modified the code

The subsequent stage is to keep in touch with some code or expansion of code or adjustment of code that makes the test pass. The new code composed at this stage is not flawless and may, for instance, breeze through the test in an inelegant way. That is adequate in light of the fact that it will be enhanced and sharpened in Cleanup the code stage. The main reason for the composed code is to finish the test. The software engineers must not compose code that is past the usefulness that the test checks.

4.2.4 Check tests from test suits

On the off chance that all test cases now pass, the developer can be certain that the new code meets the test necessities, and does not break or corrupt any current highlights or needs or prerequisites or standards. On the off chance that they dont, the new code must be balanced until the point that they do.

4.2.5 Refresh the source

Refresh the source is for to re-factor or removal of old incorrect code because there are many versions of source code. In those codes only final updated code is correct and rest of code are not as per requirement of application or software.

4.3 Tools, frameworks and other concepts:

4.3.1 Task:

Task is a collection of instructions, which is used for to do some individual job.

4.3.2 Task runners:

Task runners are the technology which is use for manages and organize the sequence of execution of tasks.

4.3.3 Jasmine:

Jasmine is a development framework ,which is behavior-driven.

4.3.4 Karma:

Karma is a tool, which is use for execution of source code like JavaScript and plotting result on browser or command prompt.

4.3.5 PhantomJS:

PhantomJS is a very lightweight procedure.

4.3.6 FOG:

FOG computing is same as cloud computing, but difference is that FOG perform all its duty on local controller and directly interact with IoT devices.

Chapter 5

Discussion of Solution

Gulp task runner, with the use of built in plugins, we can solve the problem of risk filled routine development and production in application life cycle with automated tasks such as minifying, uglifying, concatenation of files, rename, versioning, task-re-usability, source-mapping, image optimization, encryption, pipe re-usability, compression, vulcanizing and helper for minimizing the effect of error, faster cache accessing Gulp operations. Also, using jasmine, karma, and phantomJS we extend Gulp task with TDD (Test-driven development) for unit testing and solve the problem of time and cost of application development. Gulp task runner can be used in FOG and Cloud models as well and can also be applied in data analytics phase on data fetched from different IoT devices like PLCs other devices like router, modem, mobile phones, Television, set-top box, etc. Fetching the data from different IoT devices like PLCs other devices like router, modem, mobile phones, Television, set-top box and storing in the local controller server by the help of CPE WAN Management Protocol TR-069, which is application layer protocol. And using both Cloud computational model and FOG computational model with accessing of same data source. And we are focusing on WiFi band steering use case for data analytics phase of this project. Using java programming language and MySQL server 5.7, we are implementing storage of data, and analytics algorithm. Using above architecture, we can achieve speed data transfer and consistency in data.

WiFi band steering use case by this data analytics phase reduce the congestion in WiFi. therefore, Throughput of network will increase. we can say that when we reduce the congestion then failure of packet transfer is also reduced because proper bandwidth is available for that and solve the problem of improvement of throughput of network after analyzing problem of data traffic and failure issues over the network. So, we can say that using Gulp task runner, Jasmine framework, Karma tool, PhantomJS browser, application layer protocol CPE WAN Management Protocol TR-069, Java programming language, database MySQL Server 5.7, Cloud computation model and FOG computation model, we are solving all the problems which are described in problem definition.

References

[1] V. Partners, *Gulp vs. Grunt for JavaScript Task Automation.* Available at http://www.velocitypartners.net/blog/2016/02/19/gulp-vs-grunt-for-javascript-task-automation/ as on date 01/04/2018.

[2] Dennisaa, *HTML, Angular, Json.* Available at https://dennisaa.wordpress.com/author/dennisaa/page/24/ as on date 01/04/2018.

[3] K. Sitterberg, *Awesome testing with NetBeans, Angular 2 and TypeScript.* Available at https://jaxenter.com/awesome-testing-with-netbeans-angular-2-and-typescript-126078.html as on date 01/04/2018.

[4] dennisaa, *Gulp and Karma: simple example.* Available at https://dennisaa.wordpress.com/2016/05/29/gulp-and-karma-screenshots/ as on date 01/04/2018.

[5] S. Mohammad, *Testing AngularJS Service using Jasmine.* Available at http://saifikram.com/2014/04/testing-angularjs-service-using-jasmine as on date 01/04/2018.

[6] Basta, *2015 - Basta! 2015, DE: JavaScript and build.* Available at http://www.xoriant.comhttps://www.slideshare.net/lennybacon/2015-basta-2015-de-javascript-and-build as on date 01/04/2018.

[7] B. Forum, *TR-069 CPE WAN Management Protocol.* Available at https://www.broadband-forum.org/technical/download/TR-069_Amendment-5.pdf as on date 01/04/2018.

[8] B. Forum, *TR-181 Device Data Model*. Available at https://www.broadband-forum.org/technical/download/TR-181_Issue-2_Amendment-11.pdf as on date 01/04/2018.

[9] open source, *OpenACS*. Available at https://github.com/gangsun/OpenACS as on date 01/04/2018.

[10] smallnetbuilder, *Reduce Wi-Fi Congestion With Band Steering*. Available at https://www.smallnetbuilder.com/wireless/wireless-howto/32754-reduce-wi-fi-congestion-with-band-steering/ as on date 01/04/2018.

[11] T. P. Blog, *2.4GHz vs. 5GHz*. Available at http://phorus.com/blog/2.4ghz-vs.-5ghz as on date 01/04/2018.

[12] L. Systems, *WLAN Band Steering*. Available at http://www.contica.pl/gfx/contica/userfiles/_public/specyfikacje_en/techpaper/tp-wlan-band-steering-en.pdf as on date 01/04/2018.

[13] open source, *jdmp*. Available at https://github.com/jdmp/ as on date 01/04/2018.

[14] M. Shilman, *Testing Frameworks*. Available at https://stateofjs.com/2016/testing/ as on date 01/04/2018.

[15] Shidhin, *React TDD Example: Unit Testing and Building a React Component With Jest & Gulp and React Test Utils*. Available at http://www.undefinednull.com as on date 01/04/2018.

[16] open source, *Write a test*. Available at http://www.protractortest.org as on date 01/04/2018.

[17] open source, *Automate and enhance your workflow*. Available at http://www.gulpjs.com as on date 01/04/2018.

[18] npm Enterprise, *Working with private modules*. Available at http://www.npmjs.com as on date 01/04/2018.

[19] open source, *packages*. Available at http://www.bower.io as on date 01/04/2018.

[20] X. Corporation, *BIG DATA ANALYTIC*. Available at http://www.xoriant.com as on date 01/04/2018.