****

**Sri Lanka Institute Of Information Technology**

Comprehensive Design/Analysis Projects

Project Proposal

Vigraha High Volume CDR Analyzing Framework

Group no:

Name: A.G.R.Delantha

DIT : IT09012844

Supervisor name: Ms.Sujani Sumanadasa

Date of submission

Abstract

The proposed system which is ‘Vigraha High volume CDR analyzing framework’, analyze the high volume data at any given time within a short period. This Project is based on analyzing call detail records and give result according to predefined rule. Each and Every telecommunication companies have more than five event types in their mobile operator domain. When there is a transaction between those events that record must store in log files. Within a single day there are 30 Million records are store on those files. Our proposed system need to analyze these data and need to execute dynamic rules according to that.

Outcome of this project is simplified user interface for create rules and execution time and our system will analyze the data according to rule and execute the rule in relevant time. Reports are produced as final outcome.

# Introduction

## Background

Vigraha is a high volume data analyzing framework that analyzes call detail records and produce result within short period of time. A call detail record is a data record produced by a telephone exchange or other telecommunications equipment documenting the details of a phone call that passed through the facility or device. Mainly there are five event types available for call detail records. Those are Voice call, Short Message Service (SMS), General Packet radio service (GPRS), Location Based Service (LBS), Unstructured Supplementary Service Data (USSD). If there any transaction between these event types the record must store in a file. There are 30 Million records can store in files within single day.

To analyze these large amounts of data we need a proper mechanism. Suppose a particular telecommunication company wants to send a promotion SMS to customers who are using IDD at regular time. But in the current system what they have, they send SMS to all of their customers. That’s wasting resources. So we need to proper data analyzing mechanism to analyze these data and send SMS to customers who are using IDD at regular time. This is what our system does. It analyzes the data and select customers who are using IDD at past two months and send SMS to only those customers. These are called promotion programs. But in our system it has more than that. Promotion Programs, Loyalty programs, Tenure programs are some of those.

Promotional programs are based on activation, initial events or regular events. Activation can be SIM activation, GPRS activation, or MMS activation and initial events can be first SMS or first MMS at any given time and regular events can be voice calls, SMS, top-ups , LBS campaigns.

Loyalty programs are based on subscriber usage in voice calls, SMS, GPSRS, Roaming and other relevant areas. In these programs system need to run daily scan for the qualified subscribers and identify the winners at the end of specific period to send SMS to encourage them.

Tenure programs are based on long standing subscribers in SMS, SIM Reload, GPRS, number of top-ups. System needs to identify the winner at the end of the specific period and winner will notify via SMS.

We divide the system into four major independent modules. Those are,

* **Data loader**
* **Admin – UI**
* **Reporting UI**
* **Rule Engine**

**Data loader** module is responsible for decode the CDR files and store relevant fields into database. This module is an independent server always up and running but the uploading event need to happen in at any configured time.

**Admin UI** is responsible for create rules based on above mention three programs. Also it includes other basic functionalities like login, registration, Open ID integration, Central authentication service (CAS).

**Reporting UI** is responsible for generate reports at any given time. Mainly there are 30 reports need to show including drill down reports. It also has above basic functions I mentioned.

**Rule Engine** module is responsible for executing rules at any configured time. Before executing, data need to be analyzed using apache hadoop map reduce programme.

## Research Problem

The main research problem in this document is analyzing large number of data within short period of time. To analyze data we are going to use **apache hadoop** project. It’s an open source project so we need to configure it to our main objectives. Apache hadoop is a project from apache foundation and it use by Google, Facebook, Twitter, Yahoo and many other large organization for purpose of data analyzing. Apache hadoop project has two main areas. Those are Hadoop distributed file system (HDFS) and Map reduce. So we only need Map reduce but we need to do a research and developments for map reduce using a language called PIG that comes with hadoop project.

Also we have other research areas like integrate **central authentication service** (CAS) with open ID integration for user interfaces.

# Objectives of the Project

‘Vigraha’ High volume CDR analyzing framework, have both specific and general objectives.

## Specific Objectives

* Analyze large amount of data.
  + Can analyze large amount of data. This framework can be use for any telco company with smaller configurations.
* Hadoop Map Reduce
  + Use hadoop map reduce for data.

## General Objectives

* Central authentication Service with Single Sign On
* Open ID Integration
* High Availability of data
* Security
* Concurrent access
* Generate Reports

## Research Questions

1. How to handle large amount of data without affecting system performance?
2. What are the benchmark results for other databases comparing with mysql?
3. How to handle server event occurring schedules?
4. How to integrate hadoop map reduce to system?
5. How to segment subscriber based on their activities?
6. How to simulate large number of sample data without taking longer time?
7. How to tune database performing with large number of data?

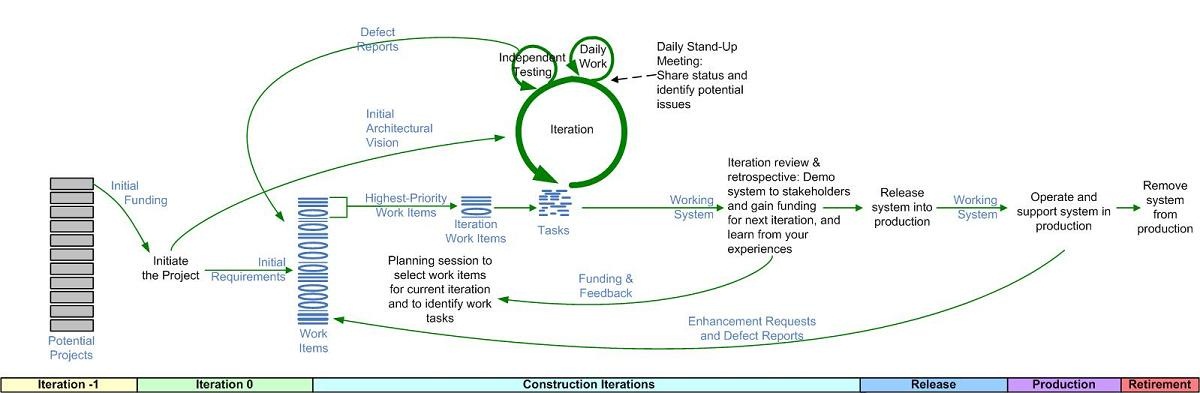
# Methodology

‘Vigraha’ High volume CDR analyzing framework is based on analyzing data and executes rules at any given time. To achieve this need to focus on research and development with initial requirements. When the project goes need to add more requirements and need to do more research and developments. There are many methodologies available today. To fulfill above features we use **Agile Methodology** and **Scrum** sub methodology.

## Scrum Methodology

Of all the agile methodologies scrum is unique because it introduced the idea of ‘empirical process control’. Scrum use real world progress of a project to plan and schedule releases. Scrum is based on sprint, which is can be two weeks to give working system to customer. In our system we use four weeks for each sprint and two sprints for a release.

**Scrum Development Cycle**



**Scrum Roles**

* **Product Owner**
  + Responsible for communicating the vision of the product to the development team.
  + Requirement prioritizing.
  + Decide when product is ready to ship.
* **Scrum Master**
  + Give hardware and software requirements.
  + Protect the team from disruption.
  + Teach scrum to team.
* **Team**
  + Self Organizing.
  + Cross functional.

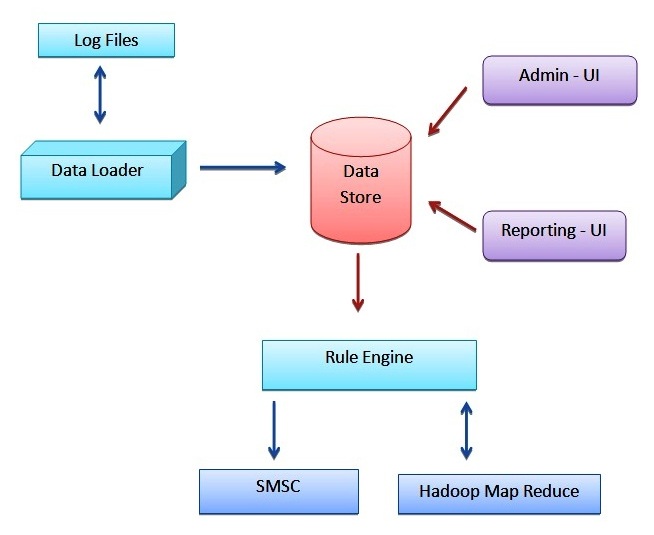
**Initial User Stories**

In scrum entire system need to describe with user stories. Following are the user stories what we identified.

1. Read log files
2. Archive Log files
3. Summery table uploading schedules
4. Spring Integration
5. Initial Services
6. CAS integration with Single Sign On
7. Open ID Integration
8. Create rules based on programs
9. Search rule
10. View and edit created rules
11. Data analyze with map reduce
12. Configure properties
13. Initial Reports
14. Identify Drill down reports and charts
15. Identify necessary indexes
16. Spring Birt integration
17. Test cases
18. Concurrent access
19. Security
20. SSL for User interfaces

Here are the initial user stories. After Identify user stories we need to estimate those above stories.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User Stories | | | | | | | | | | | | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Rajith | 3 | 3 | 5 | 8 | 13 | 8 | 5 | 3 | 3 | 3 | 13 | 5 | 8 | 5 | 3 | 13 | 5 | 0 | 0 | 0 |
| Lasantha | 3 | 3 | 3 | 8 | 13 | 5 | 8 | 5 | 3 | 3 | 13 | 5 | 5 | 5 | 5 | 13 | 5 | 0 | 0 | 0 |
| Thejani | 3 | 3 | 3 | 13 | 13 | 8 | 8 | 8 | 3 | 3 | 13 | 5 | 8 | 5 | 3 | 13 | 5 | 0 | 0 | 0 |
| Naveen | 3 | 3 | 5 | 8 | 13 | 8 | 8 | 5 | 3 | 3 | 13 | 5 | 8 | 5 | 3 | 13 | 5 | 0 | 0 | 0 |
| **Final Extimation** | 3 | 3 | 5 | 8 | 13 | 8 | 8 | 5 | 3 | 3 | 13 | 5 | 8 | 5 | 3 | 13 | 5 | 0 | 0 | 0 |

**System High Level Architecture (Will change this)**

**My Contribution**

My Contribution of the project is Rule Engine module. Rule Engine module is the one that executes rules at any given time. This module need to be up and running the entire time and fire event only for execute rules. This module need to integrate into hadoop map reduce to retrieve the analyzed data. This module use Spring, Hibernate and quartz scheduler as main technologies and use many properties to configure the system.

Database

Example Rule:

Send SMS to IDD lovers at 2012-02-20 01:00

Analyze data to retrieve IDD lovers numbers

Hadoop Map reduce

Execute Rule at 2012-02-20 01:00

Retrieve Rule to execute

Send Final Execution Result to database

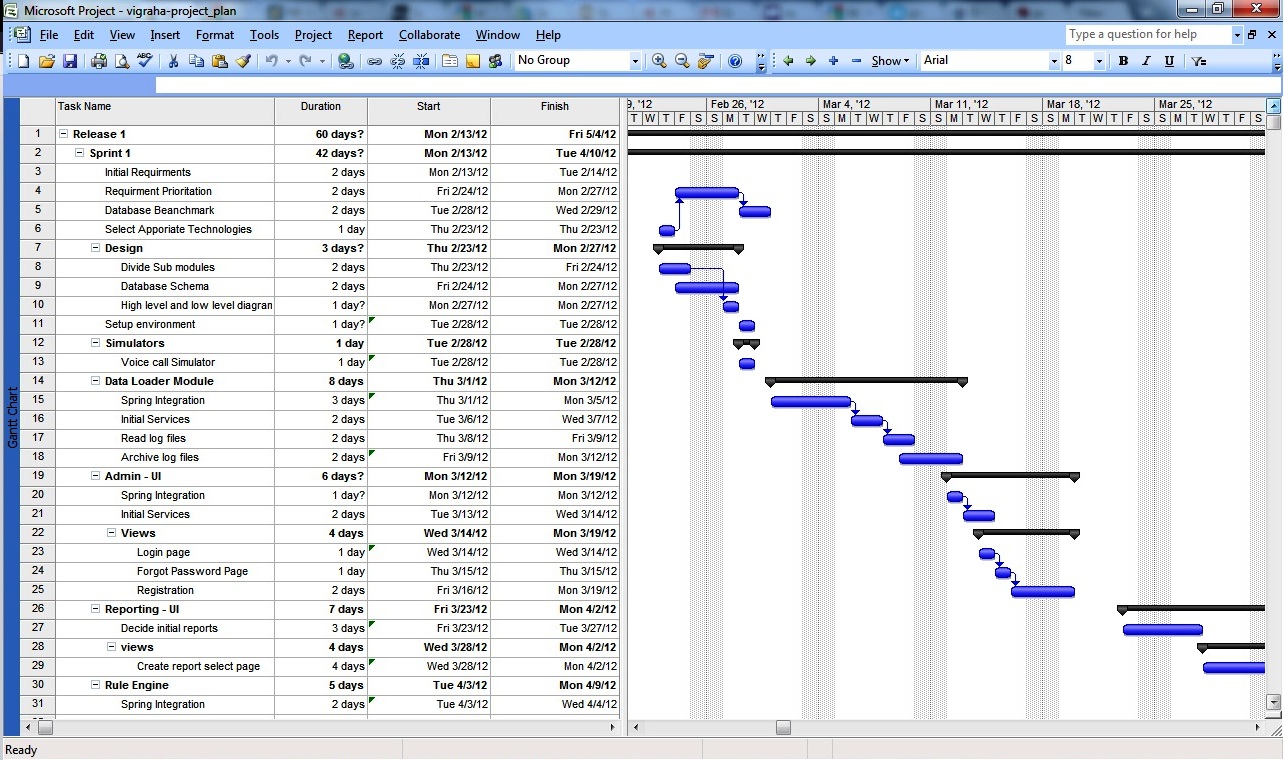
Analyze data Using Hadoop Map reduce

SMS Center

Send SMS to SMS Center

* **If the execution failed due any reason, rule engine will execute the same rule again after 10 minutes.**

First rule engine get the rule that created from the admin user interface from the database. Analyze the data according to rule and execute the rule at given time. If the execution failed it will re-execute the same rule after every 10 minutes until it successful.

**Project Plan**

Detail Project Plan : <https://github.com/rajithd/vigraha/tree/master/docs/TBS>

**Work Break Down Structure**

**Vigraha High Volume CDR Analyzer**

Analyzer

Administrator UI

Maintain log files

Report UI

Create stimulators

Send records to database

Register and log administrator

Get requirements

Generate rules

Execute rule

Analyze records

Send SMS

Get requirements

Generate report

Software Requirements for development

* Intellj Idea 11.0 version
* DB Visuualizer
* Eclipse – Use for report design

Graphics

* Microsoft project2007 - Use to develop Gant chart.
* Corel Draw 12 – Use to draw diagrams.
* Photoshop - To design the interfaces of the system

Technologies

* Apache hadoop
* Apache maven
* Apache tomcat
* Spring framework
* Quartz Schduler

# Description of Personal and Facilities

|  |  |
| --- | --- |
| **Name** | **Assigned work** |
| A.G.R.Delantha | Rule Engine Module   * Execute rule * Spring integration * Data analyze with hadoop map reduce |
| Thejani Dineshika | Admin – UI   * Login * Forgot password * Initial Services * CAS with SSO * Open ID integration |
| Lasantha Perera | Reporting UI   * CAS with SSO * Initial report * Drill down report * Open ID integration * Spring birt integration |
| Naveen Dissanayaka | Data loader   * Read log files * Archive log files * Spring integration * Executing event * Summery table uploader |

# References

[1] Craig Walls, Spring in Action

[2] Madhusudhan Konda, what’s new in Java 7 ?

Appendices

# Appendix A: List of Acronyms and Abbreviations

CDR – Caller Detail Record

SMS - Short Message Service

LBS - Location Base System

GPRS - General Packet Radio Services

USSD - Unstructured Supplementary Service Data

WBS – Work Break Structure