WeServe Project Report

Tholkappian Chidambaram[1], Venkatesh Devale[2], Sasank Matavalam[3], Matthew Zhu[4]

Software Engineering Department, San Jose State University
1 Washington Square, San Jose, CA

[1]tholkappian.chidambaram@sjsu.edu
[2]venkatesh.devale@sjsu.edu
[3]sasank.matavalam@sjsu.edu
[4]matthew.zhu@sjsu.edu

***Abstract* — This document gives an overview of the 'WeServe' project implemented for CMPE 272 - Spring 2018. The goal behind this project is to provide a platform for prospective NGOs and volunteers to easily find each other based on an effective machine learning matching algorithm. In this project, a recommendation engine was developed to perform the matching utilizing ElasticSearch and PySpark.**

***Keywords* — NGO, Volunteer, Machine Learning, Node.JS, MySQL, ReactJS, ElasticSearch**

I. Introduction

Community and social projects are undertaken by various entities as part of the legal and humanitarian obligations. Community projects cover a wide variety of domains across many different geographic regions. Managing different parties involved in these projects while fulfilling the legal requirements is a difficult task. Community projects also need to advertise the need for volunteers, reach out to community, optimal management of resources within the required timeline. Process requirements of these projects are similar even though they are implemented at different regions with different set of parties.

II. Problem Statement

In general the community projects do not have a platform to streamline the various stages of the project from start to finish. Projects lack resources to have a technology platform and apply best practices. Project managers or pro-bono consultants start out with paper based systems to collect data and make decisions. These projects also need help with consulting, outreach, collaboration, efficiency and applying proven techniques. Having a dedicated platform to support the community projects will enable the project to reach more people and will result in better social outcomes.

A. Proposal

We are proposing to build a web portal called WeServe with capabilities to support the community projects. WeServe will specifically address the problems faced by the user persona we researched, that is described in the later section. This portal will facilitate initial posting and hiring workflow for the community projects. Volunteers, NGO representatives and pro-bono consultants can work on the projects and participate in the hiring workflow according to their role. WeServe will also recommend specific projects to a volunteer and a specific volunteer for a project manager. Recommendation of the projects and volunteers will be based on a weighted score computed by a set of volunteer's attributes mentioned below.

- Volunteer and project location.
- WeServe project navigation pattern of the volunteer.
- Volunteer interests and project keywords.

B. Scope

WeServe portal will provide a basic framework for the proposal mentioned in the previous section. Portal is scoped specifically to help with initial stages of the community project and limited to hiring workflow. Portal has a capability to extend in future for the following enhancements.

1. Ongoing collaboration of the community project with volunteers and stakeholders.
2. Specific area for community projects.
3. Keyword search of specific individual or a project.
4. Web based forms instead of paper forms.
5. Project and volunteer data analytics.

C. User Persona

Community projects involve many different entities like Non-Governmental Organization (NGO), pro-bono consultants, volunteers, NGO representative, and government personnel. Some of the key project persona we observed are given below.

1) Vetrivel Foundation, *NGO*

Bio: Primary mission of Vetrivel Foundation is to bridge the digital divide in education that exists between remote villages, impoverished regions and those who have access to modern educational tools.

Goals:

- Empowering children by providing education.
- Decrease illiteracy in the world.
- Bridge the digital divide in education.

Frustrations:

- Unable to find volunteers.
- Need grass-root organizers to implement project.

2) Ram Kumar, *Student*

Bio: Ram is a student of Visual Communications in Pune, Maharashtra. He is cheerful, enthusiastic and always willing to help. He has been helping at a local school for a long time. Due to time constraint, he feels working with NGO projects will have greater reach and better utilization of his time.

Goals:

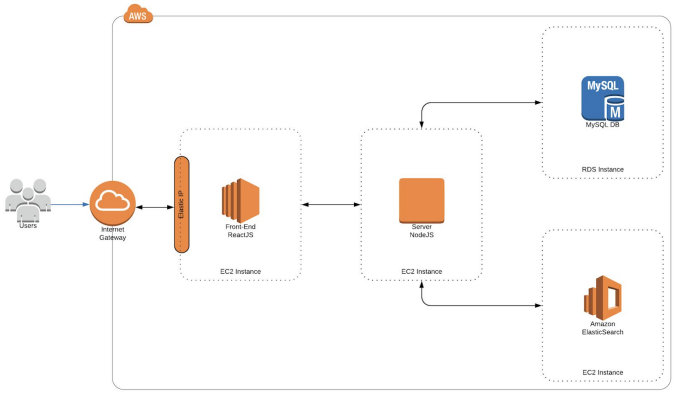
- Volunteer in social projects.
- Be a positive change in the society.
- Help the underprivileged.

Frustrations:

- Unable to find good projects.
- Resource and time constraint.
- Cannot find a good team to work with.

III. Implementation

The WeServe platform is built with the following four main components:



A. Web Front-End

We reviewed bootstrap so that we can get the design of the web front-end very attractive. We implemented the front-end in ReactJS, and hosted on AWS EC2.

ReactJS gave us the easier approach as we build custom components that we could use everywhere in the project. It was fast in rendering as it renders only the changed DOM and not the entire page itself.

It also helped us build the project as a single page application.

B. Back-End

For the back-end we have used Node.JS as our server side framework as Node.JS runs on a single thread and is a light weight process. Node.js' request-response methodology is relatively faster when compared to other server side frameworks. This back-end server is hosted on an AWS EC2 instance.

Node.JS provides a feature rich library that we can use with just importing the required modules not worrying about how they work at backend. We have used Bcrypt, express-sessions to name a few.

C. Database

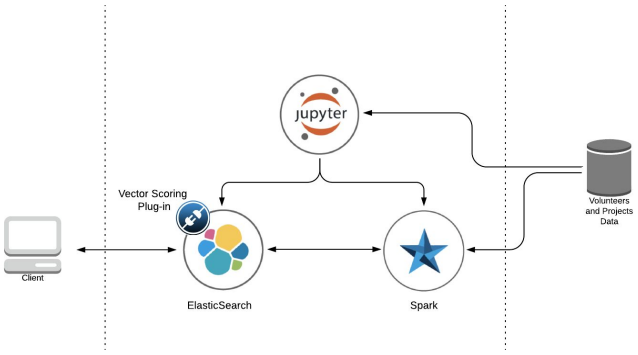
We have used MySQL database as our datasource. We used this for two reasons as we wanted relational entities mapped. For example, NGOs to Volunteers and vice versa.

MySQL also gave us advantage of easy and faster readability. We used AWS RDS to host our MySQL database.

D. Recommendation model

Recommendation is a machine learning model that is based on the concept of "wisdom of the crowd". Recommendations are based on the assumption that if 2 volunteers share similar preferences or attributes then the things that one of them prefers could be good recommendation to make to the other. For example if volunteer A tends to like certain projects, and volunteer B shares some of the preferences with volunteer A, then the projects that the volunteer A liked, that volunteer B has not yet seen, may well be projects that volunteer B will also like.

Generation of recommendations is powered by Alternating Least Square algorithm. This algorithm solves a specific type of collaborative filtering model known as matrix factorization. A matrix representation of volunteer and project data is factorized to two smaller matrices each for volunteer and project.



For WeServe, we generated random weighted score between volunteers and projects. We then trained the recommendation model in PySpark then loaded the trained model into an ElasticSearch instance running on AWS EC2. Our back-end server then queried ElasticSearch whenever we needed to display project recommendations to a volunteer or to display volunteer recommendations to an NGO.

IV. Future Enhancement

Provision for the Website Admin who can see all the posted projects. He will also get a view of the registered volunteers, NGOs. He will get a graphical views of all user session activity to see most visited projects, pages. This will enable the WeServe admins to see which category projects and NGOs are gaining help and accordingly we can come with other ideas to facilitate other NGOs and Volunteers with right consulting.

We also need to provide some sort of communication workspace where volunteers, pro-bono consultants and NGOs can share meeting details, files or any communication per se.

In the future, we plan to include the automatic generation of the weighted score between volunteer and projects based on location, similar interests, and user navigation patterns. These scores will then get injected into ElasticSearch

V. Conclusion

We found that there is currently a need for a platform where NGOs can effectively get matching volunteers for their projects without directly marketing what their needs and requirements.

Inspired from Prof. Ranjan's idea of developing this platform - 'WeServe' we tried to achieve a platform for NGOs to post their requirements as projects and based on registered volunteers, we get the matching volunteers for them which they can hire efficiently.

VI. Contributions

Tholkappian Chidambaram - Student in MS Software Engineering, SJSU. Developed the recommendation model for WeServe using IBM's ElasticSearch Spark Recommender. Designed the schema for the MySQL database to store user and project information, as well as weighted score between volunteers and projects. He also developed backup routines for MySQL, elasticsearch. He was involved in the web front-end design and created user persona definitions.

Venkatesh Devale - Student in MS Software Engineering, SJSU. Developed the back-end server for user functionality like login, registration, session handling with express-sessions, APIs for getting matching volunteers to UI, 'mark-interested' feature. Frontend functionalities like the home page, projects view, user specific views, UI routing. Volunteer hiring frontend features. Hosted the frontend and backend servers to Amazon AWS EC2 and used MySQL from AWS RDS.

Sasank Matavalam - Student in MS Software Engineering, SJSU. Developed back-end server node.js APIs for NGO, Volunteer profiles, edit profile feature, post project functionality and pagination for projects views, hired volunteers, matching projects, matching volunteers. Also developed the front-end for all these functionalities and tested it using Postman service and also provided the UI Routing and used MYSQL from AWS RDS.

Matthew Zhu - Student in MS Software Engineering, SJSU. Contributed to the understanding and implementation of the machine learning recommendation engine in ElasticSearch. Developed back-end node.js APIs to query ElasticSearch for user and project recommendations and APIs to retrieve the specified users and projects from the database. Developed front-end feature for displaying these recommendations to the users. Hosted ElasticSearch on AWS EC2.

VII. Project Artifacts

Following are the artifacts created for WeServe project:

- WeServe web platform, available at weserveyou.org
- WeServe introductory video, available at https://www.youtube.com/watch?v=BYRGomV1HDI
- WeServe project code-base, available at https://github.com/SJSU272LabSP18/Project-Team-13.
- Project-volunteer recommendation model available within the github project space.

VIII. References

[1] "Apache Spark™ - Unified Analytics Engine for Big Data." Apache Spark™ - Unified Analytics Engine for Big Data, spark.apache.org/.

[2] IBM. "IBM/Elasticsearch-Spark-Recommender." GitHub, github.com/IBM/elasticsearch-spark-recommender.

[3] MLnick. "MLnick/Elasticsearch-Vector-Scoring." GitHub, github.com/MLnick/elasticsearch-vector-scoring.

[4] "Open Source Search & Analytics · Elasticsearch." Open Source Search & Analytics · Elasticsearch, elasticsearch.org/.

[5] "Project Jupyter." Project Jupyter, jupyter.org/.

[6] "Elasticdump." Npm, www.npmjs.com/package/elasticdump.