

Purpose

The interviewers are interested in *non-coding* aspects and how I solve this problems, so I create this note and it will record my thinking processes until I solve this issue.

Before Coding - 08 Mar 2017

Tool

It is a web development that need work on both server side and client side. After reviews the requirement, I choose **Node.js** for server side. Because I have few experiences of Java/Python and no experiences of Scala/GO for the web development, mean while, my last web project is a node.js web applicaton and it gave me lots of experiences. Will use **Express** framework.

Front-end will use **Bootstrap**, **JQuery**, **AJAX** for client side operation like http requests. Will think about using **React.js** as well. It depends on the free time.

Also, after reviews the requirement, there must have a database to store vote and some related objects. Because I will change the structure of objects in database during the implementation to improve the performance, as well as SQL is not good for mutable data structure. So I will use a **NoSQL** database **MongoDB**.

So It will be **partial MEAN** web application.

Some Queries

Well, I am a little confuse about the requirements. I am thinking the purpose of this challenge.

Is it to show my web development skill or the skill to solve big data?

If it is real voting Web application ?

- I won't let logged-in/known user to vote more then 3 times. We can have a **User** database and to calculate the time of voting. If voting is more then 3 times. Then give a fail response to user. So they won't have any invalid voting at all...
- We also can have a **Candidate** database. Every valid vote will modify the candidate votes(+ 1). So CountMeUp is just very quick http response. All it will do is looking for the candidates data in database and return 5 candidates and their votes.
- According to the above assumption, the response will be given very quickly. And the challenge won't be deal with big data. But solve the problem with a large number of HTTP post requests from users in a short time. Do not have too much ideas about this right now because there has no distribute systems design and I only allow to use single system.
- I will use RESTful api for get/post http requests. I think it is a free-style web application that I can define the user input form and get output result. The only requirement is get accurate votes results and display them in the

website.

- Also there are more staff to do if it is real application. What if the max voting limitation changes? What if there are more than 5 candidates? There are more things that need to be considered...But not for this project right now.

If it is a big data challenge?

- If it is a big data skill challenge, then it should only have one object **Vote** which will include userID and candidateID in database. So the CountMeUp should iterate through a very large number of rows in a database and do the calculation in 1s.
- Well, I will think about the input data format and type. The inputs votes are in file or from database? Are they plain, JSON or XML? The inputs will be the whole votes data or input vote one by one?
- The more important is the how to design a good algorithm to get the best performance.

Estimate Final Project

According to the queries I have, I will give a web application that has both solution for a real application and big data challenge.

Server Side:

There has 3 Objects in my database:

- **Vote**(userID, candidateID),
- **User**(userID, maxVote, candidateOne, candidateTwo, candidateThree),
- **Candidate**(votes, validVotes).

Client Side:

- User Input with 2 value - userID and candidateID
- Two getVote buttons - 1. Get votes from **Candidate** Database (no calculation). 2. Get votes by iterating the **Vote** database, do the calculation in 1s and give an response.

Something Else

Is this a 3-8 hours job? Em... At least not for my project... Maybe my project can't meet the requirements well... Maybe I can't pass the technical challenge... But it is good for me and I love coding a lot. It gives me experiences. Also I had a plan to build Voting web application for a while. This could be a great start.

File I/O with big data - 10 May 2017

I just find one thing that we can have some improvement if the input data (1000,0000 votes) is stored in file. One of the biggest problem is I/O that will influence the result a lot. And one thing could improve the performance that the data is in one file or the data is in different file (like 10 files and each file have 100,0000 votes). In nodejs, we can use 'async' to handle parallelizing I/O tasks and calculate the result separately. This might improve the speed.

What will happen if what I did is not the requirements - 11 May 2017

I have spent lots of time for website development with high quality . Then I just notice that maybe the technical challenge is about dealing with big data. Tomorrow is the deadline and I don't have time to improve the performance of algorithm. I have no idea what should I do or what I can do...At least, I show the skill of full-stack web development.

Misunderstand the requirement !!!!! - 12 May 2017

I just notice I misunderstand the requirements. It is neither a web application challenge nor big data challenge.

It is just a algorithm. But we don't know the number of users.

The input is the total votes number and the percentages for different candidates. I will try my best to finish the right requirements. How can we solve it?

Misunderstand the requirement again ???

It is not like a pure algorithm challenge, because we don't know the total number of users...So the every votes should have his userID and candidateID... It is not an algorithm challenges. It should be big data challenges. Back to big data challenges again.

Finally

Well, I am totally confused about the requirements. It provides 10,000,000 votes and candidate percentages but don't tell me the number of users... It should not be the challenges of big data. I have no idea what is the requirements. It should give input data and a small example.