

Count Me Up Exercise Assignment

The aim of the task is to give you the opportunity to show your skills in web development. The specification is deliberately open for that reason. If you aren't strong in front-end development or you haven't done much server-side development, don't worry... Play to your strengths!

We are also very interested in _non-coding_ aspects and how you approach problems. Please supply any notes you make, such as showing your working, questions you might want to ask and assumptions you may have made. We also strongly recommend you use comments in any code you supply to communicate your reasoning as you work.

The languages that will be accepted are:-

- Java
- Scala
- JavaScript
- Go
- Node.js
- Python

You may also pick whichever client and server side frameworks you feel are suited to the task at hand.

Feature: counting votes

As a BBC television presenter

I want to see the counts for candidates within a given time frame

So that I can announce the winner of the competition

Scenario: partition choices sent to the queue by candidate percentage

Given 10000000 votes were received

And votes were distributed against candidates as:

candidate	percentage
candidate-1	5
candidate-2	10
candidate-3	20
candidate-4	25
candidate-5	40

And no more than 3 votes per user are allowed

When CountMeUp is asked for the results

Then it responds in under 1 seconds

And the final counts are:

candidate	count
candidate-1	500000
candidate-2	1000000
candidate-3	2000000

candidate-4	2000000	
candidate-5	3000000	

The given scenario implements the following acceptance criteria:

- Count Me Up should be accurate. So if there is a total count of 100 votes and 60% are given to candidate-1 then Count Me Up should return 60 as the count of votes for candidate-1.
- The same user can vote multiple times, up to a maximum of 3 times for the same candidate or for different ones. Count Me Up should not count a vote if the same user already exceeded the maximum allowed number of votes (that is should not count user-1 vote for candidate-5 if user-1 already voted for candidate-1, candidate-2 and candidate-3). This is the reason why candidate-5 for example received "only" 3M votes instead of 4M.
- Count Me Up should be fast. Count Me Up will be used as a close to real-time tool to constantly show the results of the competition, so it should be invoked every second or so to show progress. It follows that it should respond in less than 1 second.

Your goal is to implement count me up so that the given scenario becomes green.

What is expected

You have total freedom of implementation but a few things should appear in the final assignment:

- Please test. Feel free to add another scenario, possibly to TDD your solution.
- Please document your assumptions or your findings. It doesn't need to be a tome, just the summary of what you did, what failed, what worked and why it ended up like that.
- Your solution should run on a single machine (that is, it should not use Elastic Map Reduce and return the results back!)

What is possible but not required

- That you achieve the below 1 second performance requirement. If you are not able to squeeze it below 1 second it is relatively important. It is important that you show what steps increased the performances and what didn't work well. Of course if you can do it then it definitely shows your skills.
- That you parallelise the code over multiple cores. If you want to do it, then go for it.
- Use GIT, please consider frequent commits and speaking messages to document your implementation and the steps involved.

Submitting your test

Check your source code into a public GitHub repository and email the repository location to the contact email provided

Please include instructions with your submission or a readme.txt with details that will help us understand how to build, run & test your code

General Suggestions

- Start simple, make it work first, don't think about performances right away.
- Introduce a single coherent change at a time to be sure of the impact that change is having in isolation
- Keep an eye at the profiler to see where the application is speeding most of the time
- Be sure to measure the effect of a change that in your opinion should improve performances