Crowdsourcing, DD2471

Eric Molin, Victor Dahlin, John Storby and Kristofer Pitkäjärvi

Royal Institute of Technology KTH, Stockholm, Sweden

1 Abstract

Abstract This project have been used crowdsourcing to ease the power of CPU computation. Crowdsourcing is a method to divide a large task into smaller subtasks and invite other people to help solving these subtasks. These people will be referred to workers in this project. Given a database with a lot of different workers, specialized in different fields, the problem is to know which worker is the best for the job. It is crowd-selection's task to find the most suited worker for your problem. In this experiment we have investigated four types of approaches for selecting the best workers and finally evaluated the result. From running all four approaches we found that the third approach with lot of different variables yield the best result.

2 Introduction

2.1 What is crowdsourcing

Crowdsourcing have been used for a very long time and the idea is simple but very useful. The first known crowdsourcing event was the longitude price (1714), where the British government held a competition of whom could came up with the best idea to calculate your longitude coordinates at sea. However today most of the crowdsourcing is done for free by volounteers. With help of the Internet crowdsourcing have been even more simpler and faster to receive help from others. Students who requesting fast answer on Stackoverflow or Internet forums for people who are willing to help or solve their tasks. In science researcher uses crowdsourcing for either collect big data or sending out questionnaire to random peoples. [1]

However there has been a few different shady ways of crowdsourcing, as an example UTorrent implemented a bitcoin miner in their application and everyone who accepted their terms of services would automatically mine bitcoins for them, to excuse the bitcoin mining, they told the users that 75% of the profit were going to be donated, while the company kept 25%.[2] Another example is ESEA which is a gaming platform that had a hidden bitcoin miner The owners recieved 4000USD and kept all the profit. [3]

2.2 Project motivation

The crowdsourcing will consist of two programs, a client (worker) and a server (task manager). The task for the workers to solve will be integer factorization. We will conduct four experiments to see which method will be the fastest one of dividing tasks from the tasks mananger to the workers.

3 Related work

The authors from the paper have been working with task-driven crowd-selection and evaluated the performance of the workers. The workers was following a Bayesian generative model to decide where and what state the workers are in. The model are gathering information about the workers latent skill and latent category. The crowd manager was ranking the workers by their feedback scores and the information from the crowd database. The data set, latent category, was selected during a certain time period from Quora, Stack Overflow and Yahoo website. From Quora the data was randomly picked comparing to Yahoo and Stack Overflow where they selected the data that had the most votes or the best answers.

The authors was using Vector Space Model (VSM), Dual Role Model (DRM) and Topic Sensitive Probabilistic Model (TSPM) during the evaluating phase. VSM is based on cosine similarity by calculating the cosine angle between two vectors. DRM and TSPM are based on multinomial distribution. To summarize the TDPM was standing out most among the three other methods by high crowd-selection quality in precision and recall. [4] From this paper we are familiar with cosine similarity and will therefore use that method in our project.

4 Method

We have developed two programs, a client program and a server program. The server handles everything related to the database and manages all client connections and all tasks. The client program only deals with the connection to the server and the integer factorization itself. The client and server communicate with each other over a TCP connection, which we developed a simple protocol for to send information between the server and client. To store information on the server we use a MySQL database.

We developed and tested four different algorithms for determining which method of distributing tasks was more efficient.

4.1 Algorithm 1

The first algorithm is a simple and naive approach of simply sending the first unsolved number to the first worker, the second to the second worker and so on. Two clients will never work on the same number at the same time.

4.2 Algorithm 2

Algorithm two is based on the first algorithm, but slightly more sophisticated. This algorithm allows for two clients to work on the same number at the same time, but only if there are no other tasks available. This works by introducing a new boolean field in the database for each number to indicate that it is being worked on but not yet solved.

4.3 Algorithm 3

The third algorithm is taking a new approach and moving away from the naive approaches. When a client connects to the server, it will firstly perform a few benchmark tests which the server will use to categorize this user based on a previous baseline performance. A client will be placed in one of three different categories; 1) better performance than the baseline client, 2) similar performance as the baseline client, 3) worse performance than the baseline client. The server will then distribute tasks with an attempt to send harder tasks to better clients and easier tasks to slower clients.

4.4 Algorithm 4

Algorithm four is based on the vector space model, and each client is treated as a vector, which gets populated depending on the client's performance on the first few tasks it is sent, and then the server will send tasks which have vectors that match the server vector as close as possible. This is done by approximating cosine similarity between the vector of the task and the vector of the client.

5 Results

This is the ranking that we got when comparing total times to solve all the tasks in our database, 50 tasks of varying sizes (30-45 digits long).

Ranking:	(total time)
Algorithm 3	90%
Algorithm 4	95%
Algorithm 2	95%
Algorithm 1	100% (reference time)

6 Discussion

6.1 What algorithms were better

Out of our four approaches we found that the third one was best, the one with pre-evaluate for every client. The result of this approach depends on a lot of variables, such as, how extensive the pre-evaluation is.

The second best approach was the fourth one, the vector space model, most likely due to the size of our problem because of limited time. The vector space model requires real time evaluation after every solved task which causes a lot of load on the server. This approach is most likely only fit for big companies or research teams.

The first and second approach are very similar and will yield a very identical result due to pollard not allowing parallel computing. We saw improvements when a faster worker started working on the same task as a slower worker but that is it. If the tasks were different and allowed parallel computing the second approach would outshine the first one.

6.2 Why is this good

This is good to get the most "bang for the buck", with limited hardware and time, being able to save CPU time can be incredibly valuable, especially if it saves money on electricity too.

6.3 What can be improved

The testing data is random, since you cannot really tell how hard it is to factor a number beforehand. We classified the difficulty of the tasks by how many digits it had. We saw this classification as a big problem during our work since we could not really classify any tasks correctly, however it was a crucial part to actually be able to divide tasks among the workers.

We also believe that we would calculate the VSM-value on client side instead of server side to reduce the load on the server.

If we had more time and more resources we would also have liked to test this on many more clients and over more tasks over a much longer time, because we have not really tested the scalability of the application.

7 References

1. Crowdsourcing, Wikipedia. 2015
http://en.wikipedia.org/wiki/Crowdsourcing.
2. Engadget, Popular torrent client can steal your CPU cycles to mine bitcoins, 2015.
http://www.engadget.com/2015/03/06/utorrent-bitcoin-miner/
3. CBC, Video game league apologizes for Bitcoin scandal, 2013.
http://www.cbc.ca/news/technology/video-game-league-apologizes-for-bitcoin-scandal-1.1395323
4. Zhou Zhao, Furu Wei, Ming Zhou, Weikeng Chen, Wilfred Ng. Crowd-Selection Query Processing in Crowdsourcing Databases: A Task-Driven Approach. International Conference on Extending Database Technology, Brussels, 23-27 March 2015.