Screening exercise

Akshat Garg

March 17, 2022

QUESTION-1:- **What compiler optimization tasks do you want to tackle and why?**

1. **Filtering false warnings**:- The compiler gives a lot of false warning which are not really useful and are completely unnecessary. With the help of reinforcement learning we can just filter out/predict whether this kind of warning is really a warning or a fake warning. This will help the programmer to not unnecessary waste time and will optimize the compiler. I want to tackle this issue as I have face a lot while using gcc and g++ compilers, I got unnecessary warnings and then trying to debug the code which is completely correct , hence resulting in loss of lot of time and turns out finally it is correct

2. **Prediction of errors using pattern of code:-** The program analyzer takes a lot of time to get a particular error or warning, this part of the compiler is a very expensive operation, with Machine learning we can train the model on specific parts of code with a particular error and warning message as the output. Then, when a typical code is encountered, the ML model can immediately provide an error by examining the code's pattern.

   I want to tackle this problem as in the case of large code it takes a lot of time for the programmer to get the errors and then act accordingly, To solve this problem they will be able to work fast in a structured manner.

3. **Personalisation of compiler**:- Assume a scenario where two programmers using gcc/llvm one is embedded system engineer and other is a normal programmer. A normal programmer does not want any kind of warning and system information but an embedded system engineer wants all kinds of warnings and errors in the code. So using ML/RL we can predict what kind of code it is and give the compiler warnings/errors based on that code.
   I want to tackle this problem as I find it very fascinating designing a specific compiler accordingly.

4. **Predicting next line of code**:- The compiler can predict the next line of code before hand and do all the operations in parallel for that process and perform all the computations before hand. If the predicted nextline of code is correct then we can reduce a lot of time of computation and boost the performance of the compiler and the time taken by the compiler by a great margin.
   I want to tackle this problem as I am very familiar with the concept it uses. Normal caches also have prefetcher techniques and prefetching algorithms in which they can figure out the next memory address using the algorithm but in case of compiler we can predict the next line using ML and perform the operation. This will reduce the time taken by compiler by more than **50 percent** for decent accuracy ML algo.

5. **Other features:-**
   (a) Remove unused variables in the code.
   (b) Tying to perform more arithmetic operations then logic operations. I want to tackle this a it is a very normal issue and will boost the compiler performace by a lot.

QUESTION-2:- **Links to past coding project in Python or C++**

- Python Projects:

  1. **MULTI ORGAN CANCER DETECTION**
     (a) Worked on multiclass cancer prediction for various organs.
     (b) Used various machine learning techniques like EFFICENTNET , DNN, CNN and much more
     (c) This can also be extended to organ donation platform with doctor support
     Project Link:- LINK
  2. **SPAM DETECTION**:-
     (a) NLP on big data
     (b) Use of various machine learning algorithms like XGBoost, Decision Trees and many more
     (c) Work tokenization and lemmitization and tf-idf on big data
     (d) RAM optimization
     Project Link:- LINK
  3. **FUTURE WORD PREDICTION**
     (a) Prediction of next 4 words from given words.
     (b) LSTM is implemented from scratch
     (c) This can we further used to google search engine predictions.
     Project Link:- LINK

4. **Fastest Buzzer First**
    (a) Users get a set of general knowledge questions which they have to answer and the first person to answer 5 correct questions wins and the game ends
    (b) Socket programming
    Project Link:- LINK
5. **IMAGE EDITOR**
    (a) Changing the image properties like size, brightness convert to gray scale etc
    (b) Upload the image from computer onto the server and convert it to one of your desired options and then download it.
    (c) Django
    Project Link:- LINK

- C/C++ projects:-
    1. **Telephone Directory**:-
        (a) A digital telephone directory.
        (b) Implements all the features like add, delete,search,edit etc
        (c) Normal algorithmic thinking is used
        Project Link:- LINK
    2.
- **PPM TO GRAY**
    1. Converting ppm image to gray scale image.
    2. Basic from scratch implementation is done.
    Project Link:- LINK