cc-MATLAB Report

Name: Paul Buchana*
E-mail Address: pbuchana@andrew.cmu.edu

July 14, 2015

OS Environment: Microsoft

Programming Environment: MATLAB R2014a

Required Tool Box: Parallel Computing, for the program that computes the word frequency from a collection of tweets.

Creating Executable File:

This can be done by typing deploytool at the MATLAB prompt and intuitively following the prompts.

Running Installation: MATLAB Compiler Runtime is required.

1. Prerequisites for Deployment

. Verify the MATLAB Compiler Runtime (MCR) is installed and ensure you
have installed version 8.3 (R2014a).

. If the MCR is not installed, do the following:

(1) enter

»mcrinstaller
at MATLAB prompt. The MCRINSTALLER command displays the location of the MCR Installer.

(2) run the MCR Installer.

Or download the Windows 64-bit version of the MCR for R2014a from the MathWorks Web site by navigating to: http://www.mathworks.com/products/compiler/mcr/index.html

Appendix:

*MSECE Student: Carnegie Mellon University

Listing 1: median_unique.m Procedural Approach

```
1
2  %{
3  Author: Paul Buchana
4  E-Mail: pbuchana@andrew.cmu.edu
5  Last Modified: 14th/June/2014
6  %}
7
8  tic
9
10 cd('C:\Users\pbuchana\Dropbox\cc-MATLAB\src');
11
12 % Clearing active console, closing all figures and removing historical ↩
       list
13 % of commands.
14 clear all; close all; clc;
15
16 % Reading in the the tweets line by line from the text file.
17 fid = fopen('C:\Users\pbuchana\Dropbox\cc-MATLAB\tweet_input\tweets.txt'↩
       );
18
19 tline = fgetl(fid);
20 uniqueWordsContainer = [];
21 while ischar(tline)
22     tweetWords = textscan(tline, '%s', 'delimiter', ' ');
23     uniqueWordsCount = (length(unique(tweetWords{:})));
24     uniqueWordsContainer = [uniqueWordsContainer; uniqueWordsCount];
25     tline = fgetl(fid);
26 end
27
28 fclose(fid);
29
30 % Calculating the median number of unique words per tweet.
31 % Computing the cummulative sum of unique words per tweet and dividing ↩
       each
32 % sum by the corresponding number of tweets.
33 median = cumsum(uniqueWordsContainer)./(1:length(uniqueWordsContainer))↩
       ';
34
35 % Writting the output to a text file.
36 fileID = fopen(strcat('C:\Users\pbuchana\Dropbox\cc-MATLAB\', ...
37     'tweet_output\ft1.txt'),'w');
38 % Format specification: Microsoft Notepad requires a newline character
39 % sequence of '\r\n' instead of '\n'.
40 fprintf(fileID, '%f\r\n', median);
41 fclose(fileID);
42
43 computationTime = toc;
44
45 % Computation time output to console.
46 fprintf('Computaion Time: %f \n', computationTime);
```

Listing 2: uniqueMedian.m Functional Approach

```
1
2  function uniqueMedian()
3  %{
4  Author: Paul Buchana
5  E-Mail: pbuchana@andrew.cmu.edu
6  Last Modified: 14th/June/2014
7  %}
8
9  %Median number of unique words per tweet.
10 %   This function calculates the median number of unique words per tweet.
11 %   For each tweet, the median is updated.
12
13 tic
14
15 % Clearing active console, closing all figures and removing historical list
16 % of commands.
17 clear all; close all; clc;
18
19 % Reading in the the tweets line by line from the text file.
20 fid = fopen('C:\Users\pbuchana\Dropbox\cc-MATLAB\tweet_input\tweets.txt');
21
22 tline = fgetl(fid);
23 uniqueWordsContainer = [];
24 while ischar(tline)
25     tweetWords = textscan(tline, '%s', 'delimiter', ' ');
26     uniqueWordsCount = (length(unique(tweetWords{:})));
27     uniqueWordsContainer = [uniqueWordsContainer; uniqueWordsCount];
28     tline = fgetl(fid);
29 end
30
31 fclose(fid);
32
33 % Calculating the median number of unique words per tweet.
34 % Computing the cummulative sum of unique words per tweet and dividing each
35 % sum by the corresponding number of tweets.
36 median = cumsum(uniqueWordsContainer)./(1:length(uniqueWordsContainer))';
37
38 % Writting the output to a text file.
39 fileID = fopen(strcat('C:\Users\pbuchana\Dropbox\cc-MATLAB\', ...
40     'tweet_output\ft1.txt'),'w');
```

```
41 % Format specification: Microsoft Notepad requires a newline character
42 % sequence of '\r\n' instead of '\n'.
43 fprintf(fileID, '%f\r\n', median);
44 fclose(fileID);
45
46 computationTime = toc;
47
48 % Computation time output to console.
49 fprintf('Computaion Time: %f \n', computationTime);
50
51 end
```

Listing 3: words_tweets.m Procedural Approach

```
1
2 %{
3 Author: Paul Buchana
4 E-Mail: pbuchana@andrew.cmu.edu
5 Last Modified: 14th/June/2014
6 %}
7
8 tic
9
10 cd('C:\Users\pbuchana\Dropbox\cc-MATLAB\src');
11
12 % Clearing active console, closing all figures and removing historical ↩
    list
13 % of commands.
14 clear all; close all; clc;
15
16 % Importing the text file containing tweets.
17 fileID = fopen('C:\Users\pbuchana\Dropbox\cc-MATLAB\tweet_input\tweets.↩
    txt');
18 formatSpec = '%s';
19 tweetData = textscan(fileID, formatSpec, 'delimiter', ' ');
20
21 % Assigning tweetData{1} array to array variable tweetWords as to avoid
22 % unneccessary communication overhead within parfor loop.
23
24 tweetWords = tweetData{:};
25
26 % Extracting unique words and sorting them.
27 uniqueWords = sort(unique(tweetWords));
28
29 % Performning a word count. Using a parfor loop as to take advantage of
30 % Parallel Computing Toolbox capabilities for solving computationally ↩
    and
31 % data-intensive problems using multicore processors, GPUs, and computer
32 % clusters in case of large text file feeds.
```

```
33 wordCount = zeros(length(uniqueWords), 1);
34 parfor i = 1:length(uniqueWords)
35     % Frequency of occurence of each word.
36     count = size(find(strcmpi(uniqueWords(i), tweetWords)), 1);
37     wordCount(i) = count;
38 end
39
40 % Writting the output to a table.
41 Table = table(uniqueWords, wordCount, 'VariableNames', {'Word', ...
42     'Frequency'});
43
44 % Displaying table in the console.
45 disp(Table);
46
47 % Writting the result to a text file and removing column names.
48 writetable(Table, strcat('C:\Users\pbuchana\Dropbox\cc-MATLAB\', ...
49     'tweet_output\ft2.txt'), 'WriteVariableNames', 0, 'Delimiter', ' ');
50
51 computationTime = toc;
52
53 fprintf('Computaion Time: %f \n', computationTime);
```

Listing 4: tweetedWords.m Functional Approach

```
1
2 function tweetedWords()
3 %{
4 Author: Paul Buchana
5 E-Mail: pbuchana@andrew.cmu.edu
6 Last Modified: 14th/June/2014
7 %}
8
9 %Number of times each word has been tweeted.
10 %   This function takes in a text file containing tweets and perfoms a
11 %   count of unique words therein.
12
13 tic
14
15 % Clearing active console, closing all figures and removing historical ↩
      list
16 % of commands.
17 clear all; close all; clc;
18
19 % Importing the text file containing tweets.
20 fileID = fopen(strcat('C:\Users\pbuchana\Dropbox\cc-MATLAB\', ...
21     'tweet_input\tweets.txt'));
22 formatSpec = '%s';
23 tweetData = textscan(fileID, formatSpec, 'delimiter', ' ');
24
```

```
25 % Assigning tweetData{1} array to array variable tweetWords as to avoid
26 % unneccessary communication overhead within parfor loop.
27
28 tweetWords = tweetData{:};
29
30 % Extracting unique words and sorting them.
31 uniqueWords = sort(unique(tweetWords));
32
33 % Performning a word count. Using a parfor loop as to take advantage of
34 % Parallel Computing Toolbox capabilities for solving computationally ↩
       and
35 % data-intensive problems using multicore processors, GPUs, and computer
36 % clusters in case of large text file feeds.
37 wordCount = zeros(length(uniqueWords), 1);
38 parfor i = 1:length(uniqueWords)
39     % Frequency of occurence of each word.
40     count = size(find(strcmpi(uniqueWords(i), tweetWords)), 1);
41     wordCount(i) = count;
42 end
43
44 % Writting the output to a table.
45 Table = table(uniqueWords, wordCount, 'VariableNames', {'Word', ...
46     'Frequency'});
47
48 % Displaying table in the console.
49 disp(Table);
50
51 % Writting the result to a text file and removing column names.
52 writetable(Table, strcat('C:\Users\pbuchana\Dropbox\cc-MATLAB', ...
53     '\tweet_output\ft2.txt'), 'WriteVariableNames', 0, 'Delimiter', ' ')↩
         ;
54
55 computationTime = toc;
56
57 fprintf('Computaion Time: %f \n', computationTime);
58
59 end
```