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

```matlab

%{
Author: Paul Buchana
E-Mail: pbuchana@andrew.cmu.edu
Last Modified: 14th/June/2014
%}

tic

cd('C:\Users\pbuchana\Dropbox\cc-MATLAB\src');

% Clearing active console, closing all figures and removing historical list
% of commands.
clear all; close all; clc;

% Reading in the the tweets line by line from the text file.
fid = fopen('C:\Users\pbuchana\Dropbox\cc-MATLAB\tweet_input\tweets.txt');

tline = fgetl(fid);
uniqueWordsContainer = [];
while ischar(tline)
    tweetWords = textscan(tline, '%s', 'delimiter', ' ');
    uniqueWordsCount = (length(unique(tweetWords{:})));
    uniqueWordsContainer = [uniqueWordsContainer; uniqueWordsCount];
    tline = fgetl(fid);
end

fclose(fid);

% Calculating the median number of unique words per tweet.
% Computing the cummulative sum of unique words per tweet and dividing each
% sum by the corresponding number of tweets.
median = cumsum(uniqueWordsContainer)./(1:length(uniqueWordsContainer))';

% Writting the output to a text file.
fileID = fopen(strcat('C:\Users\pbuchana\Dropbox\cc-MATLAB\', ...
    'tweet_output\ft1.txt'),'w');
% Format specification: Microsoft Notepad requires a newline character
% sequence of '\r\n' instead of '\n'.
fprintf(fileID, '%f\r\n', median);
fclose(fileID);

computationTime = toc;

% Computation time output to console.
fprintf('Computaion Time: %f \n', computationTime);
```

Listing 2: uniqueMedian.m Functional Approach

```matlab

function uniqueMedian()
%{
Author: Paul Buchana
E-Mail: pbuchana@andrew.cmu.edu
Last Modified: 14th/June/2014
%}

%Median number of unique words per tweet.
%   This function calculates the median number of unique words per tweet.
%   For each tweet, the median is updated.

tic

% Clearing active console, closing all figures and removing historical list
% of commands.
clear all; close all; clc;

% Reading in the the tweets line by line from the text file.
fid = fopen('C:\Users\pbuchana\Dropbox\cc-MATLAB\tweet_input\tweets.txt');

tline = fgetl(fid);
uniqueWordsContainer = [];
while ischar(tline)
    tweetWords = textscan(tline, '%s', 'delimiter', ' ');
    uniqueWordsCount = (length(unique(tweetWords{:})));
    uniqueWordsContainer = [uniqueWordsContainer; uniqueWordsCount];
    tline = fgetl(fid);
end

fclose(fid);

% Calculating the median number of unique words per tweet.
% Computing the cummulative sum of unique words per tweet and dividing each
% sum by the corresponding number of tweets.
median = cumsum(uniqueWordsContainer)./(1:length(uniqueWordsContainer))';

% Writting the output to a text file.
fileID = fopen(strcat('C:\Users\pbuchana\Dropbox\cc-MATLAB\', ...
    'tweet_output\ft1.txt'),'w');
```

```
% Format specification: Microsoft Notepad requires a newline character
% sequence of '\r\n' instead of '\n'.
fprintf(fileID, '%f\r\n', median);
fclose(fileID);

computationTime = toc;

% Computation time output to console.
fprintf('Computaion Time: %f \n', computationTime);

end
```

Listing 3: words_tweets.m Procedural Approach

```

%{
Author: Paul Buchana
E-Mail: pbuchana@andrew.cmu.edu
Last Modified: 14th/June/2014
%}

tic

cd('C:\Users\pbuchana\Dropbox\cc-MATLAB\src');

% Clearing active console, closing all figures and removing historical list
% of commands.
clear all; close all; clc;

% Importing the text file containing tweets.
fileID = fopen('C:\Users\pbuchana\Dropbox\cc-MATLAB\tweet_input\tweets.txt');
formatSpec = '%s';
tweetData = textscan(fileID, formatSpec, 'delimiter', ' ');

% Assigning tweetData{1} array to array variable tweetWords as to avoid
% unneccessary communication overhead within parfor loop.

tweetWords = tweetData{:};

% Extracting unique words and sorting them.
uniqueWords = sort(unique(tweetWords));

% Performning a word count. Using a parfor loop as to take advantage of
% Parallel Computing Toolbox capabilities for solving computationally and
% data-intensive problems using multicore processors, GPUs, and computer
% clusters in case of large text file feeds.
```

```
wordCount = zeros(length(uniqueWords), 1);
parfor i = 1:length(uniqueWords)
    % Frequency of occurence of each word.
    count = size(find(strcmpi(uniqueWords(i), tweetWords)), 1);
    wordCount(i) = count;
end

% Writting the output to a table.
Table = table(uniqueWords, wordCount, 'VariableNames', {'Word', ...
    'Frequency'});

% Displaying table in the console.
disp(Table);

% Writting the result to a text file and removing column names.
writetable(Table, strcat('C:\Users\pbuchana\Dropbox\cc-MATLAB\', ...
    'tweet_output\ft2.txt'), 'WriteVariableNames', 0, 'Delimiter', ' ');

computationTime = toc;

fprintf('Computaion Time: %f \n', computationTime);
```

Listing 4: tweetedWords.m Functional Approach

```

function tweetedWords()
%{
Author: Paul Buchana
E-Mail: pbuchana@andrew.cmu.edu
Last Modified: 14th/June/2014
%}

%Number of times each word has been tweeted.
%   This function takes in a text file containing tweets and perfoms a
%   count of unique words therein.

tic

% Clearing active console, closing all figures and removing historical list
% of commands.
clear all; close all; clc;

% Importing the text file containing tweets.
fileID = fopen(strcat('C:\Users\pbuchana\Dropbox\cc-MATLAB\', ...
    'tweet_input\tweets.txt'));
formatSpec = '%s';
tweetData = textscan(fileID, formatSpec, 'delimiter', ' ');

```

```
% Assigning tweetData{1} array to array variable tweetWords as to avoid
% unneccessary communication overhead within parfor loop.

tweetWords = tweetData{:};

% Extracting unique words and sorting them.
uniqueWords = sort(unique(tweetWords));

% Performning a word count. Using a parfor loop as to take advantage of
% Parallel Computing Toolbox capabilities for solving computationally and
% data-intensive problems using multicore processors, GPUs, and computer
% clusters in case of large text file feeds.
wordCount = zeros(length(uniqueWords), 1);
parfor i = 1:length(uniqueWords)
    % Frequency of occurence of each word.
    count = size(find(strcmpi(uniqueWords(i), tweetWords)), 1);
    wordCount(i) = count;
end

% Writting the output to a table.
Table = table(uniqueWords, wordCount, 'VariableNames', {'Word', ...
    'Frequency'});

% Displaying table in the console.
disp(Table);

% Writting the result to a text file and removing column names.
writetable(Table, strcat('C:\Users\pbuchana\Dropbox\cc-MATLAB', ...
    '\tweet_output\ft2.txt'), 'WriteVariableNames', 0, 'Delimiter', ' ');

computationTime = toc;

fprintf('Computaion Time: %f \n', computationTime);

end
```