**CS 59000SA** **Fall 2018**

# Assignment 6 Simple Query using Apache Spark

*Instructor. Ninghui Li*

Due: 11:59pm Dec 7th, 2018

## 6.1 Introduction

In this assignment your are going to use Apache Spark to finish a series of tasks.

### 6.1.1 Dataset

Click the following link to download the dataset.

- **Training Dataset**
- **Testing Dataset**
- **Description of the dataset**

### 6.1.2 PySpark:

PySpark is an interface to Apache Spark implemented in Python. In this lab, you are going to use PySpark on scholar cluster or your own machine. Please check **here** for a tutorial. If you already have Spark installed on your machine, please take a look at sample codes located at : './bin/pyspark python/examples/'

## 6.2 Task 1: Basic Query

Your first task is to implement algorithms for the following queries. You should create a **separate** python file for each type of query. You must use Spark for this task.

### 6.2.1 TopK.py

This query find the top k frequent types given a attribute name, for example: proto. The attribute name can be found in the description of the dataset.
This query should take 3 arguments: fileName, attributeName, and k. It should outputs the k most frequent attribute values and their frequencies for the given attribute.
The output should be printed line by line in descending order based on frequency.
For example:

TopK.py sampleData.txt proto 2

The desired output would be:

tcp 31231
udp 21412

### 6.2.2 SpecificType.py

This query is similar to the previous one, but finds the frequency of a specific type. For example, TCP. This query should take 3 arguments: fileName, attributeName, and attributeType. It should outputs the frequency of that type. For example, SpecificType.py sampledata2 proto tcp.

To check if you got the correct frequency counts, try to reuse your lab1 code or simply use the counting feature of text editors.

## 6.3 Task 2: Conditional Queries

Conditional queries are queries with given search condition, for example, finding packages using UDP package but has connection time longer than 5000ms.
Similarly to Task 1, your need to create **separate** python files for the following objects. You must use Spark for this task as well.

### 6.3.1 Range Count Query

Range count queries are used to find items in a specific numerical range.
RangeCount.py should take 4 arguments: filename, attributeName, and 2 integers $f_1$ and $f_2$ as arguments. You can assume $f_1 < f_2$. This query counts the number of packages whose numerical value for attributeName is in the range $[f_1, f_2]$. The output should be an integer indicating how many packages are in this range.

### 6.3.2 Bicondition Query

BiCondition queries are used to find how many packages satisfying two **categorical** conditions at the same time. For example: UDP packages with CON state.
BiCondition.py should take 4 arguments, filename, attributeName1, attributeType1, attributeName2, and attributeType2 as argument. This query counts the number of packages satisfy the given conditions.
The output should be an integer indicating how many packages with such properties.

### 6.3.3 HeatMap Query

Heat map queries are used to divide packages based on two given attributes. e.g., connection time and protocols. In such case, you will print the number of packages for all combination of connection time and all protocols.
HeatMap.py takes 3 arguments: filename, attributeName1, and attributeName2.
The format of the output will be a matrix. You should print all possible values of attributeName1 in the first row and all possible vales of attributeName2 in the first column. For the remaining cells, you should

print out the frequency of packages with properties based on row and column numbers.
**For Example:**

tcp udp
0.05 5 6
0.1 21 1

### 6.3.4 Special Query

The last query find how many packages with the following property: given two numerical attribute names, the summation of those two values exceeds threshold k.
QueryForFun.py should take 4 arguments, filename, numerical attributeName1, numerical attributeName2, and k as argument. This query counts the number of packages satisfy the the following condition: attribute1 + attribute2 > k
The output should be an integer indicating how many packages satisfy the summation condition.

## 6.4 Apache Spark on Scholar Cluster:

If you intend to use Apache Spark on our cluster, please follow the guideline from **ITAP**

## 6.5 Grading:

- Queries: 90 pts (15 pts each)
- Comments: 10 pts
- Coding standard: Up to 10 points deduction. Your program should be easy to read. (Friendly variable names, sufficient comments, well structured, and etc)