COMP3210/6210 - Big Data
Assignment 1

Semester 1, 2022
Macquarie University, School of Computing

Due: At the end of recess, Friday 22 April, 5pm
Demonstration and Marking: during Week 8, in Practical sessions

Weighting: 20%

In this assignment you will:

- acquire hands-on experience in designing, implementing and querying a NoSQL database.
- implement MapReduce techniques for the processing of Big Data. You will build your assignment on top of Hadoop (i.e. an open-source version of MapReduce written in Java).

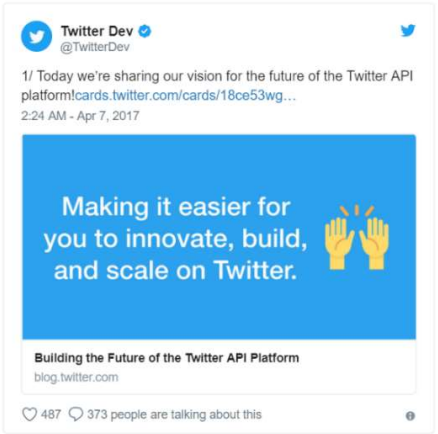
Note: If you cannot setup Hadoop on your local system, you can use online services such as:
https://studio3t.com/

This Assessment Task relates to the following Learning Outcomes:
- Apply techniques for storing large volumes of data.
- Apply Map-reduce techniques to a number of problems that involve Big Data.

Dataset

Twitter[1] serves many objects as JSON[2], including *Tweets and Users*. These objects all encapsulate core attributes that describe the object. Each Tweet has an author, a message, a unique ID, a timestamp of when it was posted, and sometimes geo metadata shared by the user. Each User has a Twitter name, an ID, a number of followers, and most often an account bio. With each Tweet, Twitter generates 'entity' objects, which are arrays of common Tweet contents such as hashtags, mentions, media, and links. If there are links, the JSON payload can also provide metadata such as the fully unwound URL and the webpage's title and description.

So, in addition to the text content itself, a Tweet can have over 140 attributes associated with it. Let's start with an example Tweet:



The following JSON illustrates the structure for these objects and *some* of their attributes:

[1] https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json

[2] JSON is based on key-value pairs, with named attributes and associated values. These attributes, and their state, are used to describe objects.

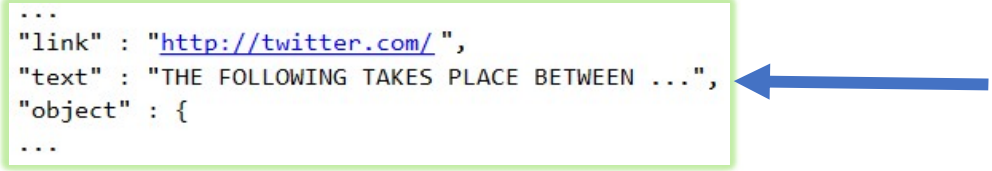

```
{
 "tweet": {
  "created_at": "Thu Apr 06 15:24:15 +0000 2017",
  "id_str": "850006245121695744",
  "text": "1/ Today we're sharing our vision for the future of the Twitter API platform!https://cards.twitter.com/cards/18ce53wgo4h/3xo1c … ",
  "user": {
   "id": 2244994945,
   "name": "Twitter Dev",
   "screen_name": "TwitterDev",
   "location": "Internet",
   "url": "https:\/\/dev.twitter.com\/",
   "description": "Your official source for Twitter Platform news, updates & events. Need technical help? Visit https:\/\/twittercommunity.com\/ \u2328\ufe0f #TapIntoTwitter"
  },
  "place": {

  },
  "entities": {
   "hashtags": [

    ],
    "urls": [
     {
      "url": "https:\/\/t.co\/XweGngmxlP",
      "unwound": {
       "url": "https:\/\/cards.twitter.com\/cards\/18ce53wgo4h\/3xo1c",
       "title": "Building the Future of the Twitter API Platform"
      }
     }
    ],
    "user_mentions": [

    ]
  }
 }
}
```


Task 1 (20 %)

- Dataset: 10000 Tweets; dataset on iLearn "tweets.zip"
- Task: Write a program in Python, to read the Tweet *Dataset* from MongoDB. Then for each Tweet, extract keywords from the text of the Tweet. Then for each Tweet, add a new name/value pair to store the keywords in a comma-separated value (CSV) format; and update the original Tweet in the MongoDB.

Task 2: (10%)

- Dataset: 10000 Tweets; dataset on iLearn "tweets.zip"
- MapReduce: Calculate the count of number of occurrences of each word in the text of Tweets.
- Create a short documentation in which you briefly describe your implementation:
  - What to write in the mapper(s) ? Flowchart and Pseudocode !
  - What to write in the reducer(s) ? Flowchart and Pseudocode !

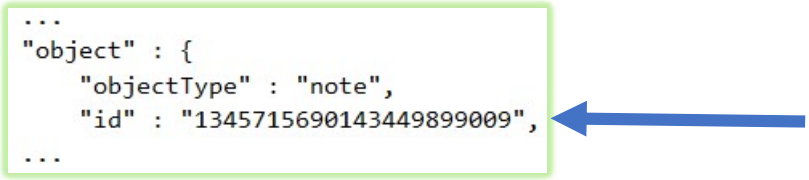


Task 3: (10%)

- Dataset: 10000 Tweets; dataset on iLearn "tweets.zip"
- MapReduce: Calculate the count of number of tweets for a list of different cities in Australia.
- Create a short documentation in which you briefly describe your implementation:
  - What to write in the mapper(s) ? Flowchart and Pseudocode !
  - What to write in the reducer(s) ? Flowchart and Pseudocode !

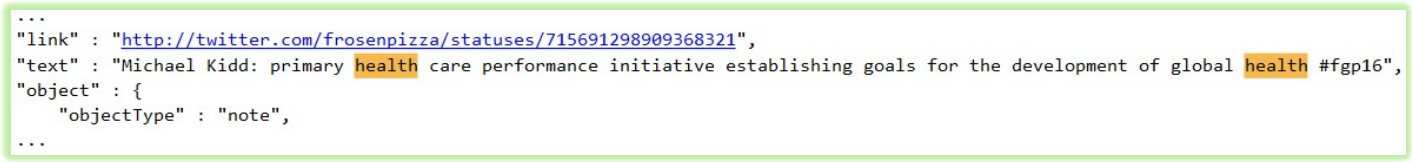
Task 4: (30%)

- Dataset: 10000 Tweets; dataset on iLearn "tweets.zip"
- MapReduce: Implement the **Merge Sort**[3] algorithm using Map-Reduce.
- MapReduce: Implement the **Bucket Sort**[4] algorithm using Map-Reduce.
- Create a short documentation in which you briefly describe your implementation:
  - How many MapReduce Jobs? Why?
  - What to write in the mapper(s) ? Flowchart and Pseudocode !
  - What to write in the reducer(s) ? Flowchart and Pseudocode !
- Sort Tweets, using the object.id :



Task 5: (30%)

- Dataset: 10000 Tweets; dataset on iLearn "tweets.zip"
- MapReduce: Implement the **TF-IDF** algorithm using Map-Reduce for the term "health" in the text of the Tweets.
- Create a short documentation in which you briefly describe your implementation:
  - How many MapReduce Jobs? Why?
  - What to write in the mapper(s) ? Flowchart and Pseudocode !
  - What to write in the reducer(s) ? Flowchart and Pseudocode !



[3] https://en.wikipedia.org/wiki/Merge_sort
[4] https://en.wikipedia.org/wiki/Bucket_sort

Submission:

Submit a zip file including:

- A documentation for each task including the Flowchart and Pseudocode
- Source code for the mapper(s) and reducer(s)
- Output for each task

Marking Rubric:

Task	Modules	Mark	
Task 1	Module 1: Read the Tweet *Dataset* from MongoDB	5	20
	Module 2: Keyword extraction from the text of the Tweet	5	
	Module 3: add a new name/value pair to store the keywords in a comma-separated value (CSV) format	5	
	Module 4: update the original Tweet in the MongoDB	5	
Task 2	Module 1: mapper(s)	5	10
	Module 2: reducer(s)	5	
Task 3	Module 1: mapper(s)	5	10
	Module 2: reducer(s)	5	
Task 4	How many MapReduce Jobs? Why?	10	30
	Module 1: mapper(s)	10	
	Module 2: reducer(s)	10	
Task 5	How many MapReduce Jobs? Why?	10	30
	Module 1: mapper(s)	10	
	Module 2: reducer(s)	10	
		Total	100

