Assignment 2

FIT 5148 - Distributed Databases and Big Data

Due: Week 12 Monday 5 PM

Worth: 25% of final marks

Background

Previously, our data analysts helped Monash University on their campaign to predict and stop fire in Victorian cities. We employed parallel techniques that we learnt in this unit to analyse their data and provide them with results. During the project handover, our data analysts suggested them MongoDB as an alternative to store their sensor data. They took the suggestion but need our expertise again with MongoDB and Spark streaming to help them analyse their data. We are required to build an application, a complete set up from streaming to storing and analyzing the data for them.

What you are provided with

- Four data sets:
    - Fire data-Part1
    - Fire data-Part2
    - Weather data-Part1
    - Weather data-Part2
- These files are available in moodle under Assignment 2.

Programming Tasks

Task A. Querying MongoDB using Mongo shell

Use the MongoDB shell to complete the following tasks:

1. Import the data (Fire data-Part 1 and Weather data-Part 1) into two different collections in MongoDB.
2. Find climate data on *15th December 2017*.

3. Find the *latitude*, *longitude* and *confidence* when the surface temperature (°C) was between *65 °C* and *100 °C*.
4. Find *surface temperature (°C), air temperature (°C), relative humidity* and *maximum wind speed* on 15th and 16th of December 2017.
5. Find *datetime, air temperature (°C), surface temperature (°C)* and *confidence* when the *confidence* is between 80 and 100.
6. Find top 10 records with highest *surface temperature (°C).*
7. Find the number of fire in each day. You are required to only display *total number of fire* and *the date* in the output.
8. Find the *average surface temperature (°C)* for each day. You are required to only display *average surface temperature (°C)* and *the date* in the output.

Combine all the answers of Task A1-A8 into a single JavaScript file called as2TaskA.js. Make sure you test the JavaScript runs correctly. Captured the output of the JavaScript to a file called **as2TaskAOut.txt**

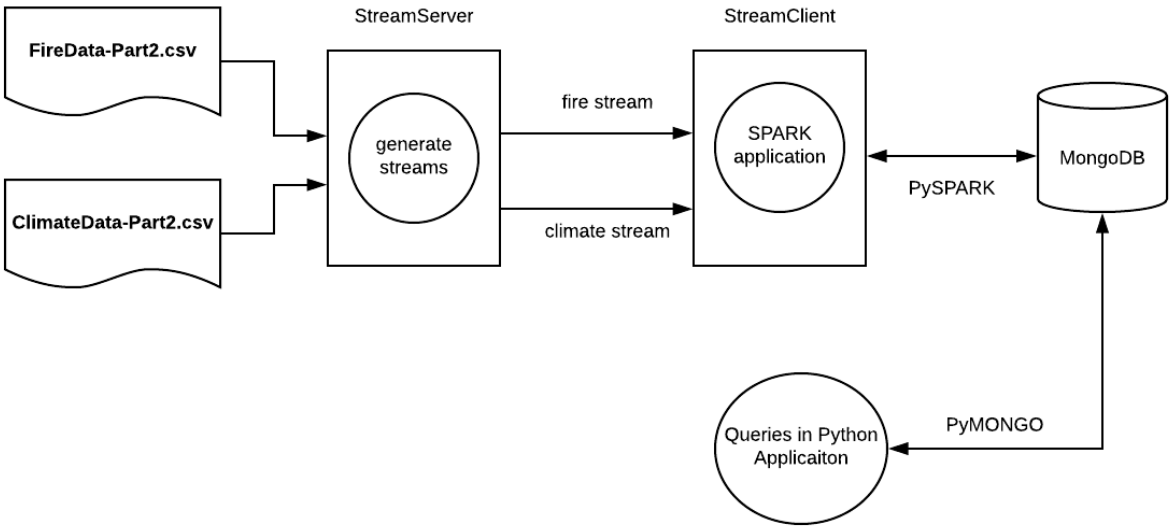
Task B. MongoDB Data Model

1. Based on the two data sets provided i.e. Fire data-Part 1 and Weather data-Part 1, design a suitable data model to support efficient querying of the two data sets in MongoDB. Justify your data model design.
2. Create a new database in MongoDB. The new database will be based on the document model you have designed in Task B1.
3. Write a python program that will read the data from Fire data-Part1 and Climate data - Part1 and load them to the new database created in Task B2.
4. Write queries to answer the Tasks A2-A8 on the new database. You need to write the queries as a python program using pymongo library in Jupyter Notebook. Please choose ONE of the Task A2-A8 and implement it in parallel.

Combine all your answers in this section into a single Jupyter Notebook file called **as2TaskB.ipynb**.

Task C. Processing Data Stream

Use the following architecture to complete Task C.



1. Write a python “generate streams” application to feed data from a file into a stream. The application should set the timing such that one tuple from “Climate Data-Part2” and five tuple from “Fire Data-Part2” are sent into the stream every second. “DataType” column in the record can be used to distinguish between the streams.
2. Write a streaming application in Apache Spark with local Streaming Context with two execution threads and a batch interval of five seconds. The application will
    a. Receive streaming data from the streams generated by the stream server.
    b. Process the streams to ensure the data can be uploaded into the MongoDB according to the data model you have designed in Task B1.
    c. Upload the result of processing in Task C2b to MongoDB.
3. Testing: Use the python application from Task B (**as2TaskB.ipynb**) to show the streamed data have been added into the database successfully.

Write Task C1 in a Jupyter Notebook called **as2TaskC1.ipynb**.
Write Task C2 in a Jupyter Notebook called **as2TaskC2.ipynb**

Marking Criteria

Credit: If you are aiming for the credit in this assignment, you need to complete all tasks described in Task A.

Distinction: If you are aiming for the distinction in this assignment, you need to complete all tasks described in Task A and B.

High Distinction Task: If you are aiming for the high distinction in this assignment, you need to complete all tasks described in Task A, B and C.

The above marking criteria suggests the **maximum grade** possible when the task is attempted. The final grade may be lower depending on:

- The correctness of the implementation,
- Program design
- Interview result (for D and HD grade).

The final grade of your assignment will be provided as an individual grade. Individual's grade may vary depending on the **contribution level** evidence by activities in the Google drive, **tasks distribution** (if any) and the result of **interview**.

Google Drive

You will be provided with a Google drive space as collaboration space for the team. You must show us a clear progression of your project throughout the development process. Label each version clearly with the last saved date, e.g. as2TaskC1-v.1-20-Mar.ipynb. Your tutor will keep track of the activity in the google drive. The activity (contribution) counts towards your assignment marks. We will not assess your Moodle submission if there is no evidence of activity in your team Google drive.

Submission

Your team should submit their final version of the assignment solution online via Moodle; you must submit the following:

- A zip file of your Assignment 2 folder, named based on your authcate name (e.g. psan002_mar005). This should contain:
    - **as2TaskA.js**
    - **as2TaskAOut.txt**
    - **as2TaskB.ipynb**
    - **as2TaskC1.ipynb**
    - **as2TaskC2.ipynb**
    - **filled contribution declaration form**.

    This should be a ZIP file and *not any other kind of compressed folder (e.g. .rar, .7zip, .tar).*
- The same assignment submission should be uploaded by EACH of the team members and must be finalised by **Week 12 Monday May 21st 2018, 5:00 PM (Local Campus Time)**. *Please note: your entire team needs to accept the assignment submission statement individually on Moodle.*
- Your assignment will be accessed based on the contents of the Assignment 2 folder you have submitted via Moodle. You should ensure that the copy of the assignment submitted to Moodle is the final version that exists in your Google Drive. We will use the same FIT servers as provided to you when marking your assignments.

Assignment Code Interview

“Interview is only required if you are aiming for D or HD in the assignment.”

During Week 12 tutorial your tutor will spend few minutes interviewing each team member to individually gauge the student’s personal understanding of your Assignment 2 code. The purpose of this is to ensure that each member of the team has contributed to the assignment and understands the code submitted by the team in their name. The final marks of the assignment will be impacted by the interview session if you are unable to explain your team’s submission. Tutors will basically focus on three questions

- One question from the part you have completed
- One question from the part you team mate has completed
- One random question (if required)

Please note: Tutorial and interview sessions will be running in parallel.

Resources

FireData: https://sentinel.ga.gov.au/#/
WeatherData: http://www.bom.gov.au/vic/?ref=hdr
Edward, Shakuntala Gupta, and Navin Sabharwal. *Practical MongoDB: Architecting, Developing, and Administering MongoDB*. Apress, 2015.
https://docs.mongodb.com/tutorials/

Other Information

How to collaborate effectively for this assignment

To effectively work in a team for this assignment, we suggest the following process be adopted by the group.

1. Establish the first meeting to plan for the project. The plan includes:
    a. an agreed regular time for a meeting,
    b. an agreed task level to complete,
    c. an agreed breakdown of the tasks and allocation
    d. an agreed deadlines of the task completion by member.

    Have all of these in writing and place the document in the Google drive. Every member is accountable for the agreed decision made on the document.

    Place an entry on an online calendar system, e.g. Google calendar on the agreed meeting dates/time and tasks deadline.

    The meeting can be conducted face-to-face or using conferencing technology such as Zoom. Monash provides access to Zoom for all students. https://monash.zoom.us/.

2. Every member needs to be able to understand and be able to write the code of all the submitted tasks. If you decide the break the task and requires individual to complete certain task, have a plan on:
    a. Overview of the overall program/code design.
    b. The data model that will be used by the program (in particular if you aim for D level or higher),
    c. Possible reusable components of the code. If there is, agree on this and write this code first so every member can use it for different tasks.
    d. How to manage the versioning of codes or components in your project.

3. Create a separate file for each task, eg taskA.1_v1_20-Mar.ipyb/js. Attempt each task in a separate workbook. Upload the individual task regularly to Google drive. Follow the agreed version naming to avoid confusion. Your tutor will check this during marking to see the progression of your project.

    Complete the task each week after you complete the tutorial exercises for the topic. Don't leave it too close to the submission deadline. If you can't meet the deadline agreed in point 1 for the task allocated to you, discuss it with your team member as soon as you realise of the problem. Don't leave it until the deadline to tell your partner that you can't complete the task. If you can't do the task, seek help early.

4. Use the regular meeting to:
    a. Check on the project progress.
    b. Consolidate the completed individual tasks to the project. This includes explaining your code to your team member.
    c. Resolve any possible confusion/problem.

    Once the meeting time is agreed as in point 1, don't change it unless it is really unavoidable. Be punctual in attending the meeting. It is disrespectful to your team member if you come late or inform your partner you can't make it in the last minute.

5. Once all the tasks are completed, combine all the codes into the notebook provided for submission. Place this final version in Google drive and name it clearly so all members can find it for the final submission in Moodle.

Having a plan, being organised and be committed to the agreed plan will increase your chance to work well with your partner.

Where to get help

You can ask questions about the assignment on the Assignment Discussion Forum on the unit's Moodle page. This is the preferred venue for assignment clarification-type questions. You should check this forum (and the News forum) regularly, as the responses of the teaching staff are "official" and can constitute amendments or additions to the assignment specification.

Also, you can visit the consultation sessions if the problem and the confusion is still not solved.

Plagiarism and collusion

Plagiarism and collusion are serious academic offenses at Monash University. Students must not share their team's work with any student outside of their team. Students should consult the policy linked below for more information.

https://www.monash.edu/students/academic/policies/academic-integrity

See also the video linked on the Moodle page under the Assignment block.

Students involved in collusion or plagiarism will be subject to disciplinary penalties, which can include:

- The work not being assessed
- A zero grade for the unit
- Suspension from the University
- Exclusion from the University

Late submissions

Submission must be made by the due date otherwise penalties will be enforced. You must negotiate any extensions formally with your campus unit lecturer via the in-semester special consideration process:

http://www.monash.edu.au/exams/special-consideration.html

There is a **10% penalty per day including weekends** for the late submission.