THE UNIVERSITY OF
SYDNEY

School of Information Technologies
Faculty of Engineering & IT

ASSIGNMENT/PROJECT COVERSHEET - GROUP ASSESSMENT

Unit of Study: COMP5349 Cloud Computing

Assignment name: Assignment 2: Apache Spark Programming

Tutorial time: 4pm-6pm THU **Tutor name:** Andrian Yang

DECLARATION
We the undersigned declare that we have read and understood the *University of Sydney Academic Dishonesty and Plagiarism in Coursework Policy*, an, and except where specifically acknowledged, the work contained in this assignment/project is our own work, and has not been copied from other sources or been previously submitted for award or assessment.

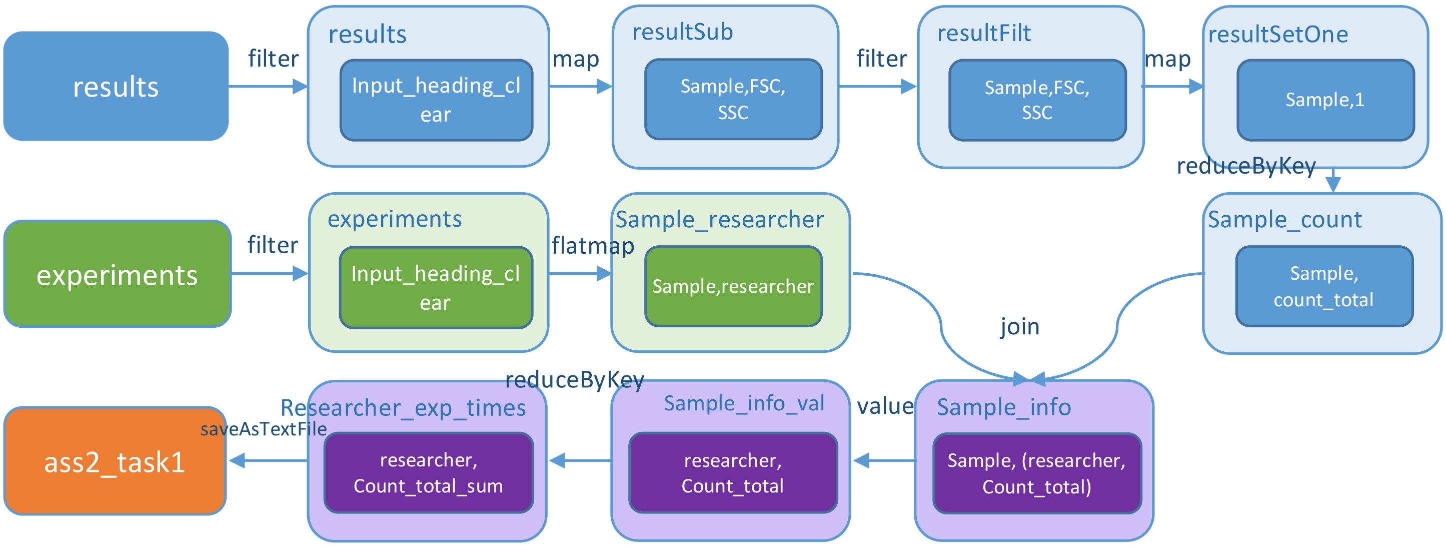
We understand that failure to comply with the *Academic Dishonesty and Plagiarism in Coursework Policy* can lead to severe penalties as outlined under Chapter 8 of the *University of Sydney By-Law 1999* (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

We realise that we may be asked to identify those portions of the work contributed by each of us and required to demonstrate our individual knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

Project team members

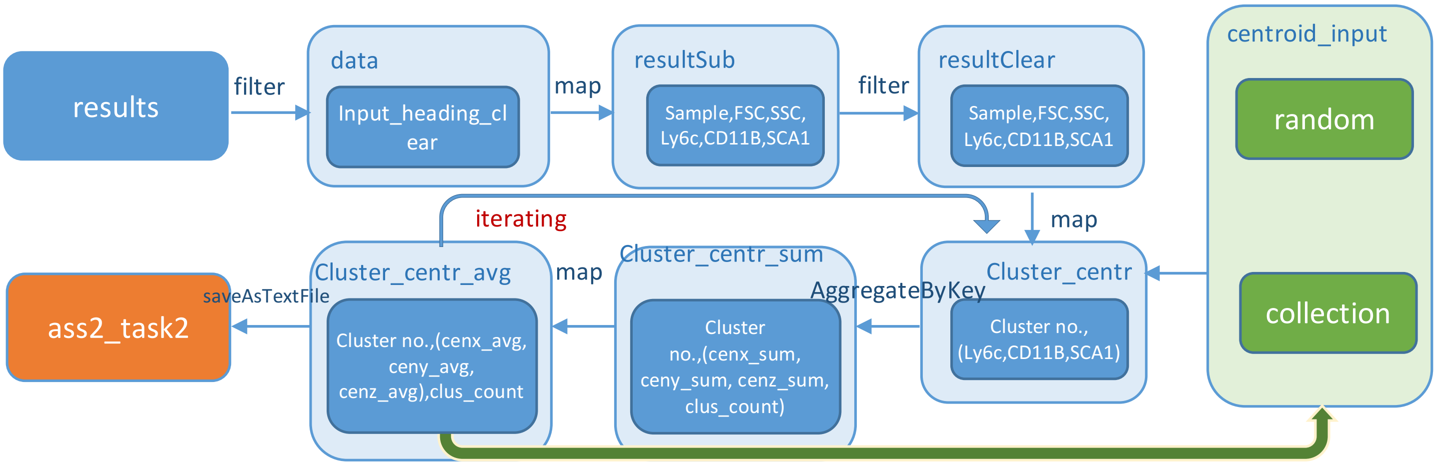
Student name	Student ID	Participated	Agree to share	Signature
1. Chengxiao Zhang	460140120	[✓ Yes] / No	[✓ Yes]/No	Cheng
2. ZhengWei Yao	450642432	[✓ Yes] / No	[✓ Yes] / No	Yao
3.		Yes / No	Yes / No	
4.		Yes / No	Yes / No	
5.		Yes / No	Yes / No	
6.		Yes / No	Yes / No	
7.		Yes / No	Yes / No	
8.		Yes / No	Yes / No	
9.		Yes / No	Yes / No	
10.		Yes / No	Yes / No	

Task 1



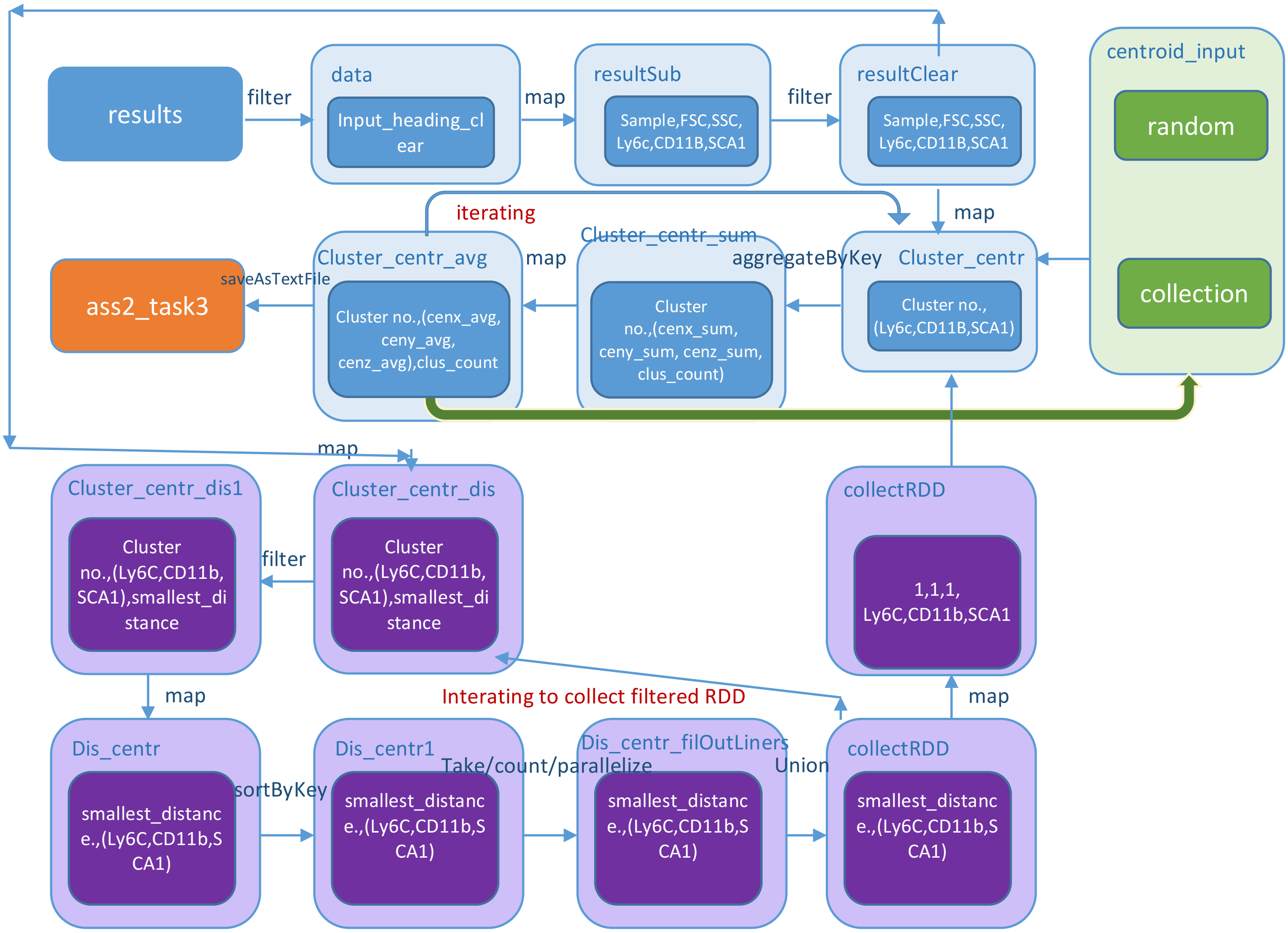
Job No.	Job name	Functions
1	results	Clear the headings row. Get a new RDD
2	resultSub	Transform into a RDD with (sample,FSC,SSC)
3	resultFilt	Filter out those that are out of range (1,150000) for both FSC and SSC
4	resultSetOne	Transform into a RDD with sample as key, and value marked as int(1)
5	Sample_count	Reduce by key, so same samples counting up
6	experiments	Clear the headings row. Get a new RDD
7	Sample_researcher	Flatmapping to separate researchers for a single key(sample)
8	Sample_info	Join two RDDs as (sample,(researcher,count_total))
9	Sample_info_val	Remove the key, making researching as a key
10	Researcher_exp_times	Reducebykey, so same research gets its total sum of experiments

Task 2


results
filter
data
Input_heading_clear
map
resultSub
Sample,FSC,SSC, Ly6c,CD11B,SCA1
filter
resultClear
Sample,FSC,SSC, Ly6c,CD11B,SCA1
centroid_input
random
collection
map
iterating
Cluster_centr_avg
Cluster no.,(cenx_avg, ceny_avg, cenz_avg),clus_count
saveAsTextFile
ass2_task2
map
Cluster_centr_sum
Cluster no.,(cenx_sum, ceny_sum, cenz_sum, clus_count)
AggregateByKey
Cluster_centr
Cluster no., (Ly6c,CD11B,SCA1)

Job No.	Job name	Functions
1	data	Clear the headings row. Get a new RDD
2	resultSub	Transform into a RDD with (sample,FSC,SSC,Ly6c,CD11B,SCA1)
3	resultClear	Filter out those that are out of range (1,150000) for both FSC and SSC
4	Cluster_centr	Input a centroid to calculate the distances, sort its minium value, assign to the cluster accordingly
5	Cluster_centr_sum	Combing aggregateByKey and functions to output sum of its centroid for x,y,z dimensions. And also count up its number of the same cluster
6	Cluster_centr_avg	Transform into averages accordingly

Task 3


results
filter
data
Input_heading_clear
map
resultSub
Sample,FSC,SSC,Ly6c,CD11B,SCA1
filter
resultClear
Sample,FSC,SSC,Ly6c,CD11B,SCA1
centroid_input
random
collection
iterating
map
Cluster_centr_sum
aggregateByKey
Cluster_centr
Cluster no.,(Ly6c,CD11B,SCA1)
Cluster_centr_avg
map
saveAsTextFile
ass2_task3
Cluster no.,(cenx_avg, ceny_avg, cenz_avg),clus_count
Cluster no.,(cenx_sum, ceny_sum, cenz_sum, clus_count)
map
Cluster_centr_dis1
Cluster no.,(Ly6C,CD11b, SCA1),smallest_distance
filter
Cluster_centr_dis
Cluster no.,(Ly6C,CD11b, SCA1),smallest_distance
collectRDD
1,1,1, Ly6C,CD11b,SCA1
map
Interating to collect filtered RDD
map
Dis_centr
smallest_distance.,(Ly6C,CD11b,SCA1)
sortByKey
Dis_centr1
smallest_distance.,(Ly6C,CD11b,SCA1)
Take/count/parallelize
Dis_centr_filOutLiners
smallest_distance.,(Ly6C,CD11b,SCA1)
Union
collectRDD
smallest_distance.,(Ly6C,CD11b,SCA1)

Job No.	Job name	Functions
7	Cluster_centr_dis	Transform into a RDD with smallest distance after the 10th interation
8	Cluster_centr_dis1	In a for loop, fiter out by cluster no.
9	Dis_centr	Put smallest_distance for each point as key
10	Dis_centr1	Sorting key ascendingly
11	Dis_centr_filOutLiners	Use take()/count()/to get 90% data and parallize it into RDD again
12	collectRDD	Union all those RDDs with same cluster.no and finally get a complete dataset without outliners. Then transform again to the right struction to compute

Appendix:

Task1 output: /user/czha0172/ass2_task1/part-00000
Task2 output: /user/czha0172/ass2_task2/part-00000
Task3 output: /user/czha0172/ass2_task3/part-00000