

# **BSDS ASSIGNMENT 4**

GROUP MEMBERS: SHUJIAN WEN, WENBO WANG, LIANG ZHANG

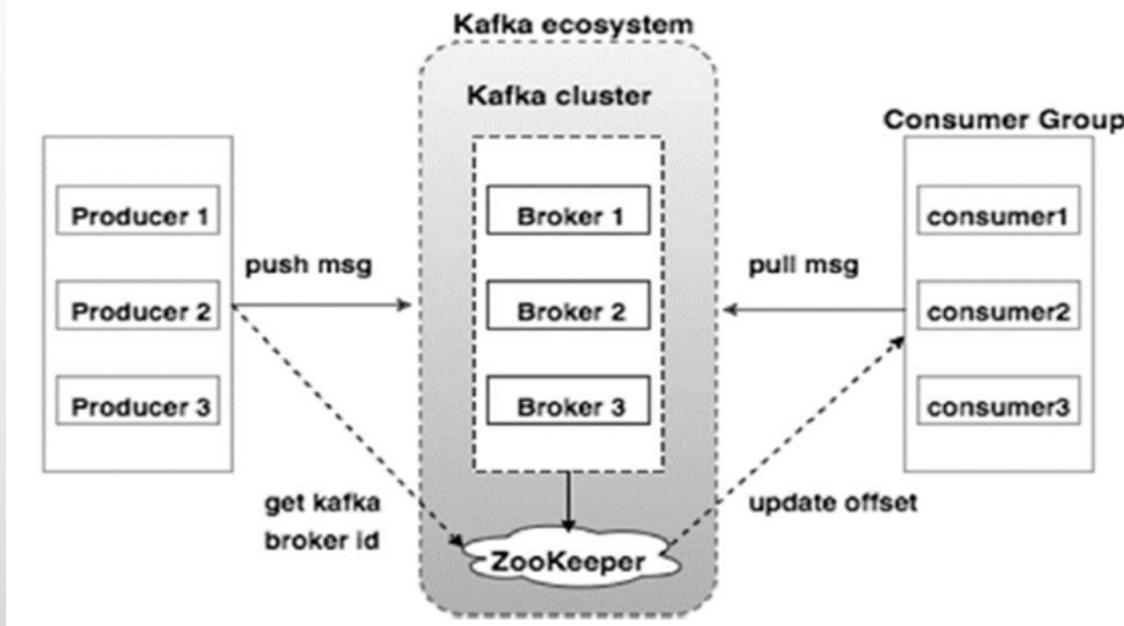
GROUP NAME: 6666

# HYPOTHESIS

- USE KAFKA TO ACHIEVE ASYNCHRONOUS APPLICATION DESIGN
- PRESUMABLY HIGHER CLIENT LOADS AND FASTER PROCESSING TIME
  - SCALABLE ARCHITECTURE
  - STREAMING API



# KAFKA ARCHITECTURE



Producer: clients POST request  
Consumer: database insertion  
Broker: topic with partitions  
Zookeeper: coordination for brokers  
(Kafka cluster)

# MEASUREMENTS

- HOW MANY CLIENTS CAN IT HOLD (32, 64, 128, 256, 512...)
- FOR EACH CLIENT, HOW FAST CAN IT PROCESS (COMPARE WITH AWS, GCP, LAMBDA)
- MORE POSTS (WITH/WITHOUT KAFKA), MORE SERVER

# METHODS

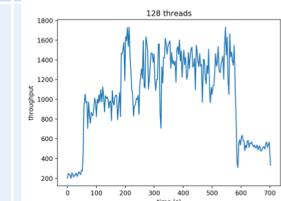
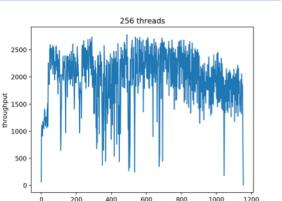
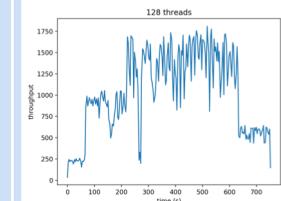
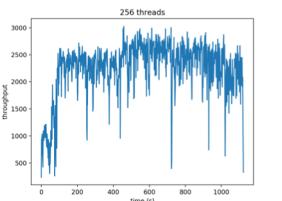
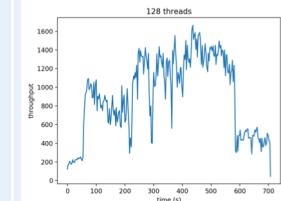
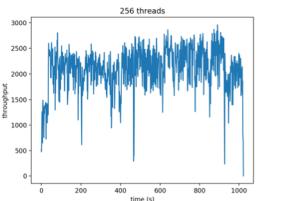
- PARAMETERS:
  - NUMBER OF CLUSTERS IN KAFKA (CLUSTER)
    - 0, 2, 4
  - NUMBER OF NODES IN EACH CLUSTER (NODES)
    - 5, 10, 15
  - POST/GET PERCENTAGE (PRODUCER)
    - 1/5, 1, 5

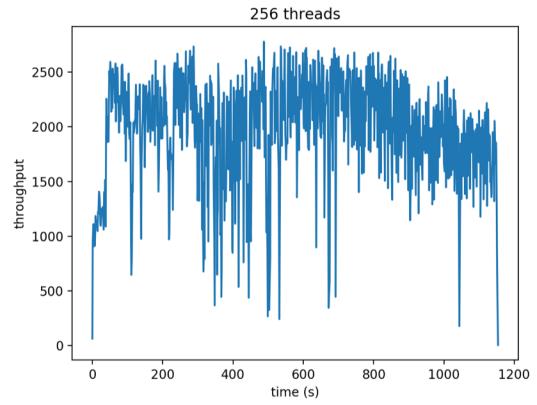
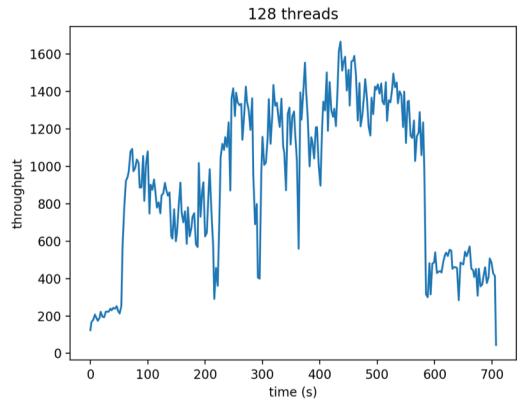
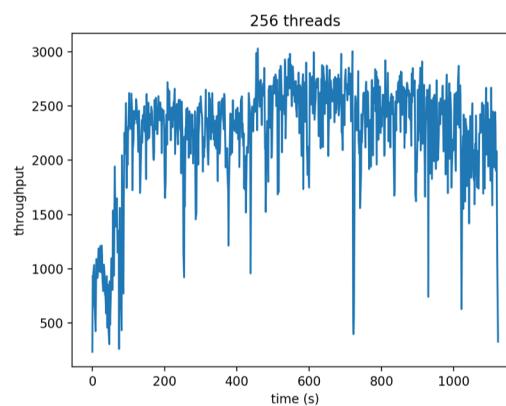
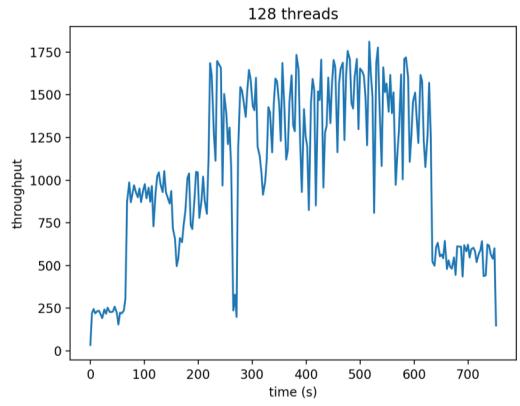
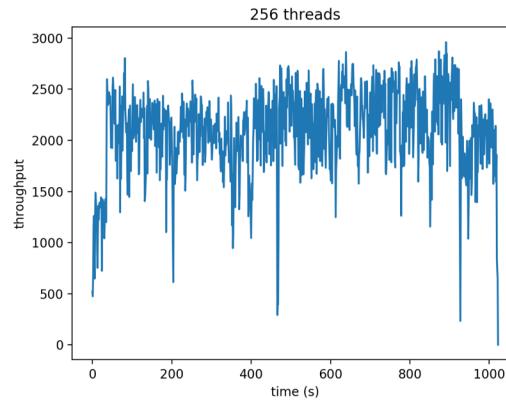
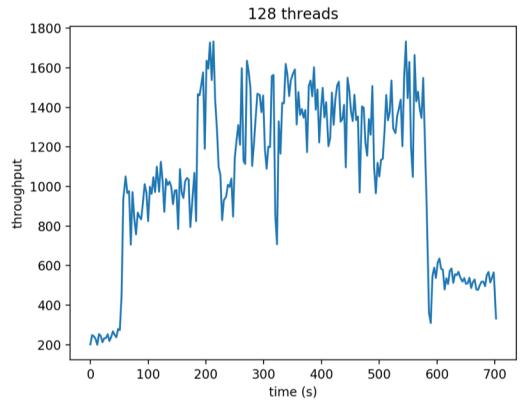
# NUMBER OF CLUSTERS IN KAFKA

WITH:

NUMBER OF NODES = 10

POST/GET PERCENTAGE = 1

Number of clusters	time					throughput	
	32	64	128	256	128	256	
0	525	556	702	1153			
2	518	562	720	1123			
4	510	548	705	1112			

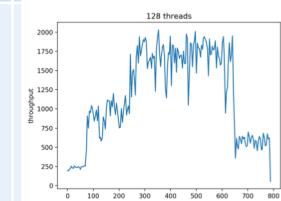
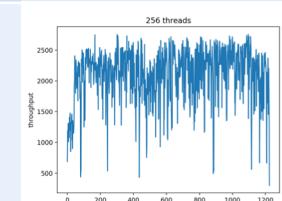
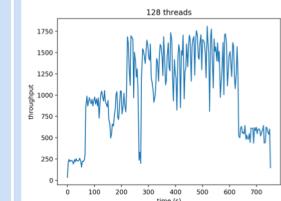
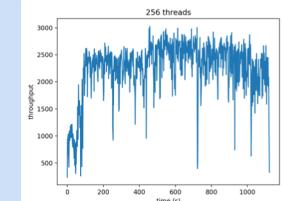
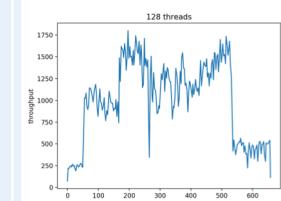
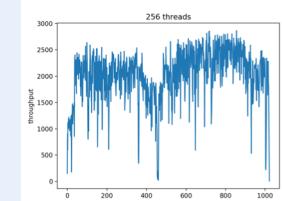


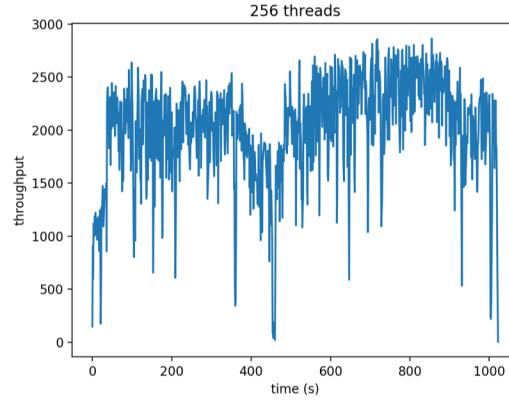
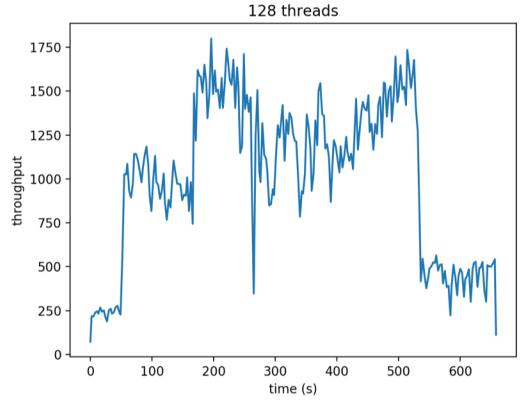
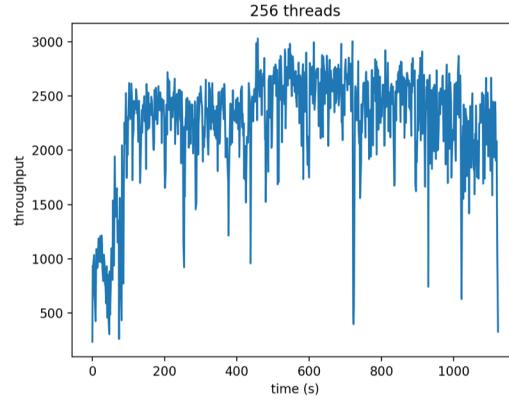
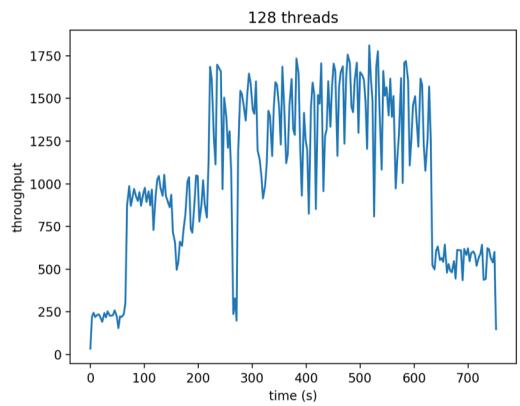
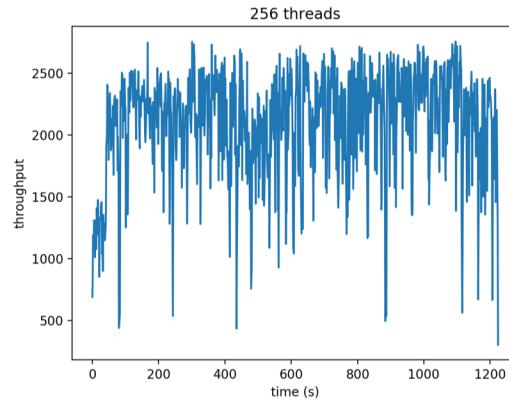
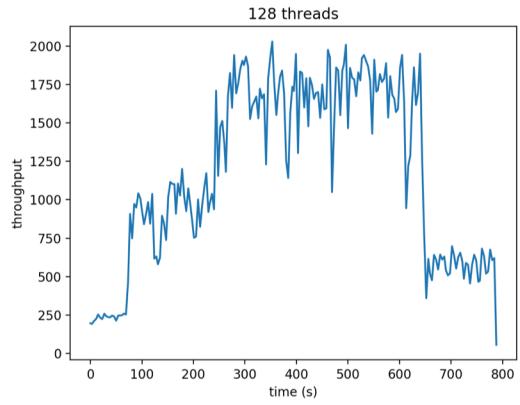
# NUMBER OF NODES IN EACH CLUSTER

WITH:

NUMBER OF CLUSTERS = 2

POST/GET PERCENTAGE = 1

Number of nodes	time					throughput	
	32	64	128	256	128	256	
5	572	621	788	1226			
10	518	562	720	1123			
15	478	514	656	1021			

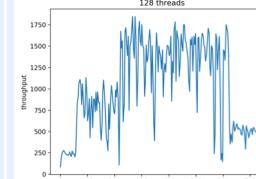
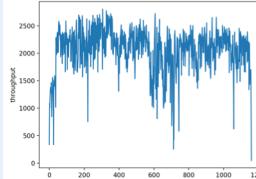
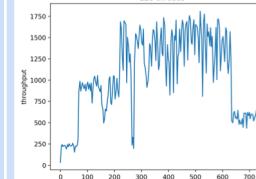
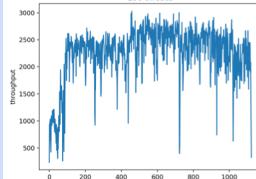
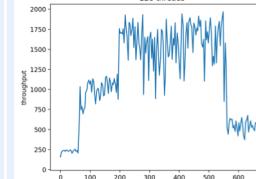
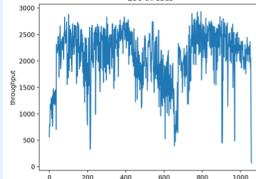


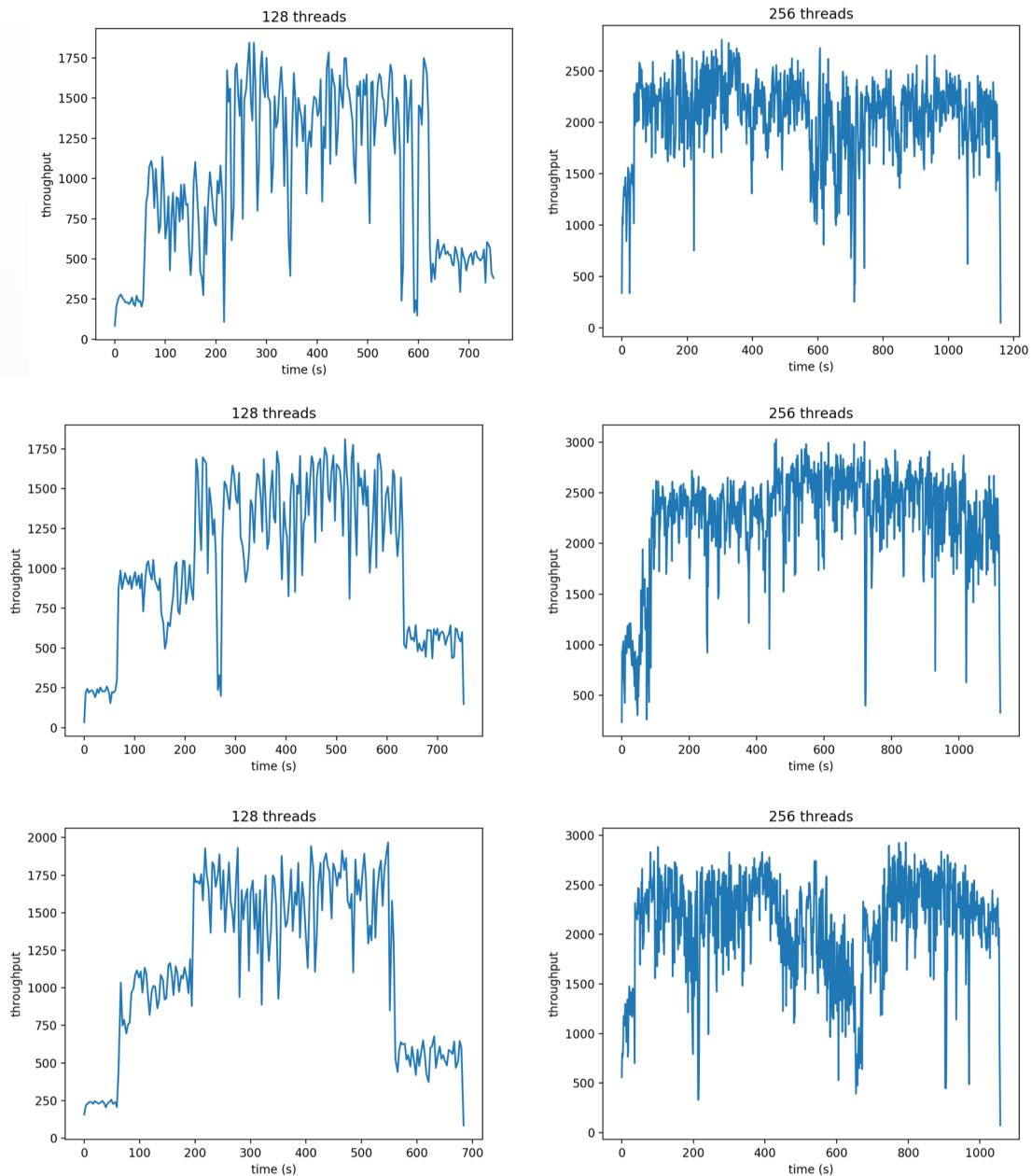
# POST/GET PERCENTAGE

WITH:

NUMBER OF CLUSTERS = 2

NUMBER OF NODES = 10

POST/GET Percentage	time					throughput	
	32	64	128	256	128	256	
1/5	548	592	749	1159			
1	518	562	720	1123			
5	498	542	684	1057			



# CONCLUSION

- ALL THE ABOVE THREE METHODS DON'T HAVE EFFECTS IN HOLDING MORE CLIENTS
  - MAXIMUM 256 CLIENTS, REACHING THE THROUGHPUT LIMIT OF 2500
  - THE LIMIT MAY DUE TO NETWORK SPEED
  - MAY RUN CLIENT ON AWS FOR FUTURE WORK
- FOR THE TOTAL RUNNING TIME
  - INCREASING NUMBER OF CLUSTERS HAS NO EFFECT
  - INCREASING NUMBER OF NODES IN EACH CLUSTER SHORTENS BY 10%
  - INCREASING POST/GET PERCENTAGE FROM 1 TO 5 SHORTENS BY 5%

[https://github.com/Greatjian/CS\\_6650\\_distributed\\_system](https://github.com/Greatjian/CS_6650_distributed_system)

**THANKS !**