**<Last Name>, <First Name>**

<Date>

Harvard Extension - Big Data Principles e88

Homework 8: Kafka

Please identify which problems were completed. If any were incomplete, please identify where you encountered problems.

|  |
| --- |
| Problem 1:  Problem 2:  Problem 3:  Problem 4:  Bonus Problem A:  Bonus Problem B: |

**Problem 1: Simple Kafka Setup [20 points]**

Show main steps and config settings you had to do to setup your 3-node Kafka cluster; show commands you used to start Zookeeper, Kafka cluster and create a new topic; and a few lines of the output on the console [5 points]

|  |
| --- |
|  |

Show a screenshot of your Kafka cluster state - output of the kafka-topics --describe command [5 points]

|  |
| --- |
|  |

Demo the console producer/consumer work [5 points]

|  |
| --- |
|  |

Show output of the DumpLogSegments tool for the 'problem1' topic [5 points]

|  |
| --- |
|  |

**Problem 2: Flume + Kafka integration [25 points]**

Paste your Flume config file below [5 point]

|  |
| --- |
|  |

Demo that your Kafka console consumer now receives events from your Web server's log in the 'problem2' topic; show content of the web server logs - and corresponding events in the Kafka topic [10 point]

|  |
| --- |
|  |

Show changes in your Flume config file to add UUID interceptor [5 points]

|  |
| --- |
|  |

Demo that events in the 'problem2' topic now have UUIDs - how can you show that? [5 points]

|  |
| --- |
|  |

**Problem 3: Programming with Kafka APIs [30 points]**

Paste your Problem3Producer code below [5 points]

|  |
| --- |
|  |

Show how you run your producer and demo the results of its work by using the Kafka console consumer [5 points]

|  |
| --- |
|  |

Paste your Problem3Consumer code below [5 points]

|  |
| --- |
|  |

Show commands you use to run your consumer and demo how it works - receiving events sent by your Problem3Producer and printing out offsets, partitions, headers and bodies of the events [10 points]

|  |
| --- |
|  |

Demo that the events sent by your producer are fairly distributed across all partitions of the 'problem3' topic [5 points]

|  |
| --- |
|  |

**Problem 4: Kafka consumer groups [25 points]**

Demo running **two** instances of your Problem3Consumer, with the same consumer group ID, listening to the 'problem3' topic with 3 partitions - which instances are processing which partitions? [10 points]

|  |
| --- |
|  |

Demo running **four** instances of your Problem3Consumer, with the same consumer group ID, listening to the 'problem3' topic with 3 partitions - which instances are processing which partitions? Are all instances processing data? Explain the results [10 points]

|  |
| --- |
|  |

Demo re-balancing of the consumers again, after bringing down two instances of the Problem3Consumer [5 points]

|  |
| --- |
|  |

**Bonus Problem A: partitioning by userID [+10 points]**

Paste your ProblemAProducer code below

|  |
| --- |
|  |

Demo its work by running the Problem3Consumer - show that events with the same userID are received from the same partition

|  |
| --- |
|  |

**Bonus Problem B: ISR or not ISR? [+15 points]**

Show status of the cluster by describing all topics; show state of the new 'problemb' topic - which nodes are leaders for which partitions?

|  |
| --- |
|  |

Show commands you used to bring down Kafka on one node. Demo the results - show how ISRs are changing for the 'problemb' topic; are you still able to send/receive events via console producer/consumers ? Demo these operations

|  |
| --- |
|  |

Demo effects of bringing the 'failed' Kafka node back up - how long did it take to get ISRs back in sync?

|  |
| --- |
|  |