Big Data Technologies

Agenda

- Kafka
- Spark Kafka Integration
- Spark Streaming

Apache Kafka

- Application1 --> Distributed Message Broker --> Application2

Uber Big Data Architecture

- https://www.uber.com/en-IN/blog/uber-big-data-platform/

Apache Zookeeper

- Zookeeper is distributed coordination and synchronization service.
- Mainly Zookeeper maintains all the metadata.

Using Kafka

- Refer screenshots

Kafka Python Clients

- step 1

```
pip3 install kafka-python
```

- step 2: create new topic

```
kafka-topics.bat --zookeeper localhost:2181 --create --topic iot --replication-factor 1 --partitions 2
```

- step 3: Implement kafka producer in python
  - Real-world application: Sensor --> Python App --> Kafka Topic
  - Demo: Python App (Random values) --> Kafka Topic
- step 4: Testing producer

```
kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic iot
```

- step 5: Implement kafka consumer in python
  - Demo: Kafka Topic --> Python App (& print)

Kafka Architecture

- App1 --> Kafka --> App2
- Kafka cluster --> Multiple nodes --> Running broker (JVM process).
- Kafka Topic --> Logical entity on which messages are sent (published) and received (subscribed).
- Topic is divided into multiple parts --> Partitions (Physical entity).
  - Each broker have one or partitions of same or different topics.
  - Data is appended into the partitions with serial numbers -- offsets.
- Topics are replicated i.e. each partition copied maintained on multiple brokers.
  - One replica -- Leader
  - Other replicas -- Followers (Replicas)
  - Replicas try to be in sync with Leader.
- Producer --> Topic partitions --> Leader
  - Round robin or Hash of key.
- Consumer <-- Topic partitions <-- Leader or Replicas.

- Consumer group -- set of consumers
  - Message is received by any one consumer in the group.

Kafka Spark Integration

- Get kafka jars (from http://172.18.4.112/kafka-jars.zip) and Add into site-packages/pyspark/jars directory to enable "kafka" source and sink support.
- Steps to run:
  - terminal1> python3 kafka-producer.py
  - terminal2> python3 iot-data-processing.py #-- may run from pycharm as well
  - terminal3> kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic avgiot