



Big Data Engineer Bootcamp

Code 4



Agenda

- **Dev Environment**
- Work with Spark
- Work with Redis
- Work with Node.js
- Interview Tips

Github Link

- <https://github.com/UncleBarney/big-data-bootcamp>



Start Docker Environment (MacOS, *nix)

- Have a docker-machine vm called bigdata
- Start a Zookeeper Container
 - `docker run -d -p 2181:2181 -p 2888:2888 -p 3888:3888 --name zookeeper confluent/zookeeper`
- Start a Kafka Container
 - `docker run -d -p 9092:9092 -e KAFKA_ADVERTISED_HOST_NAME=`docker-machine ip bigdata` -e KAFKA_ADVERTISED_PORT=9092 --name kafka --link zookeeper:zookeeper confluent/kafka`
 - If backtick is not working for you, use your virtual machine ip directly
- Start a Redis Container
 - `docker run -d -p 6379:6379 --name redis redis:alpine`



Start Docker Environment (Windows)

- Have a docker-machine vm called bigdata
- Start a Zookeeper Container
 - `docker run -d -p 2181:2181 -p 2888:2888 -p 3888:3888 --name zookeeper confluent/zookeeper`
- Start a Kafka Container
 - `docker run -d -p 9092:9092 -e KAFKA_ADVERTISED_HOST_NAME=`docker-machine ip bigdata` -e KAFKA_ADVERTISED_PORT=9092 --name kafka --link zookeeper:zookeeper confluent/kafka`
 - If backtick is not working for you, use your virtual machine ip directly
- Start a Redis Container
 - `docker run -d -p 6379:6379 --name redis redis:alpine`





Agenda

- Dev Environment
- **Work with Spark**
- Work with Redis
- Work with Node.js
- Interview Tips

Functionality

- Stream data from Kafka
 - Should be able to read from any kafka cluster
 - Should be able to read from any kafka topic
- Perform Computation
 - Average every 5 seconds
- Write back to Kafka
 - Should be able to write to any kafka cluster
 - Should be able to write to any kafka topic

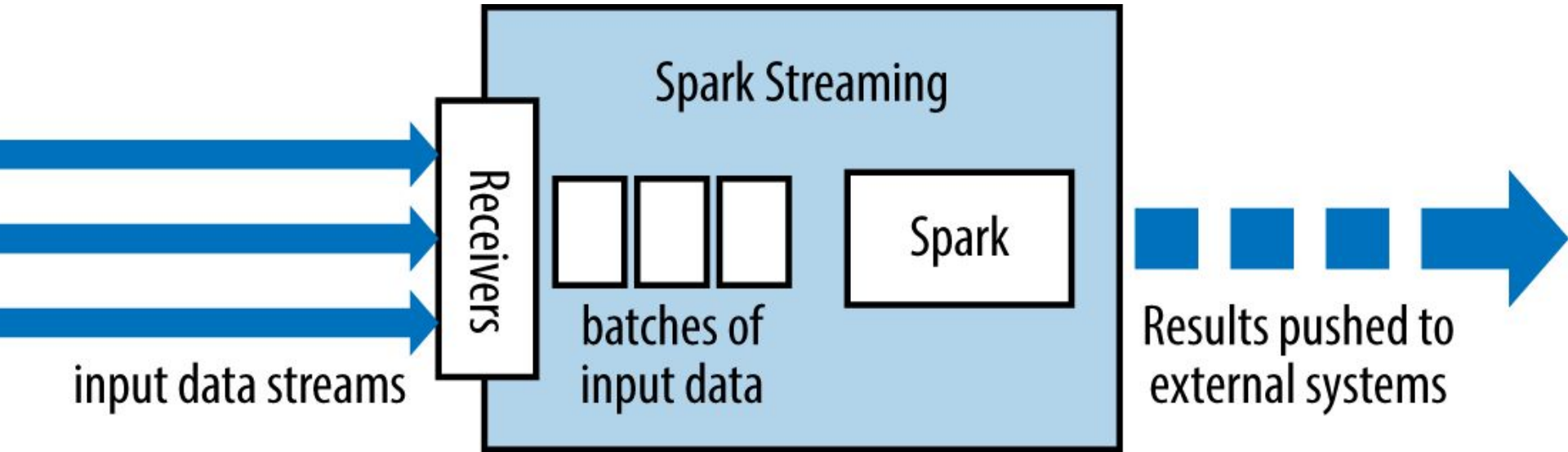


Work with Spark Using Python

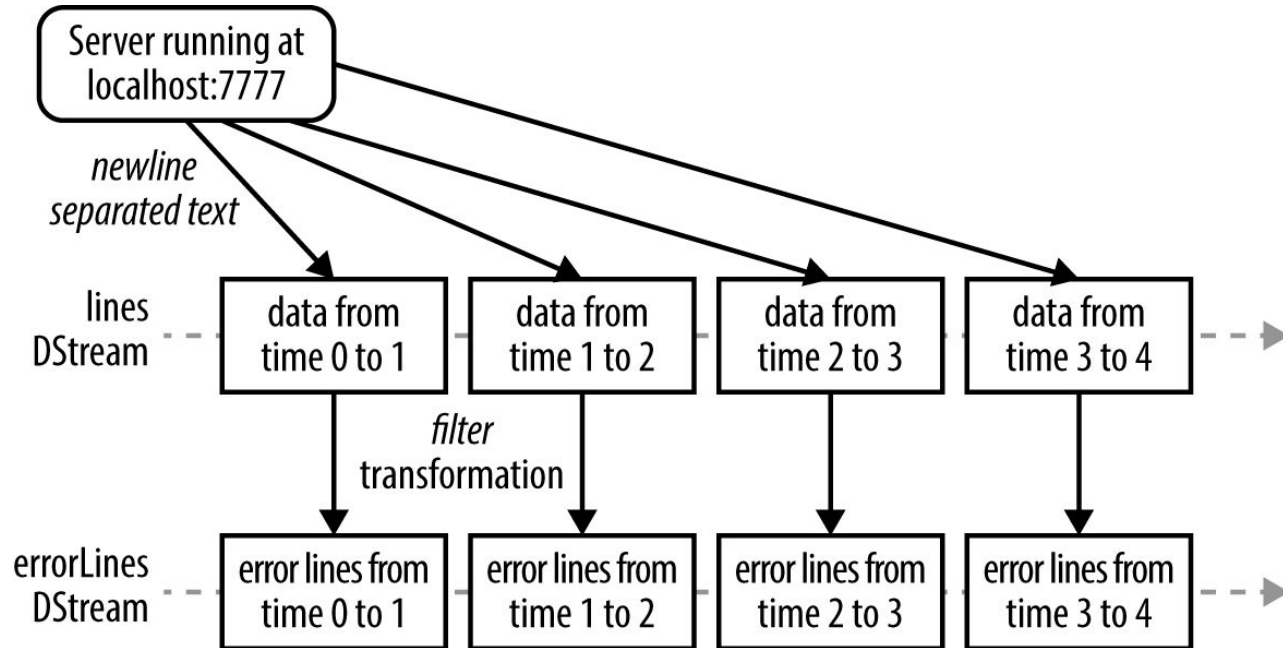
- pyspark



Spark Streaming

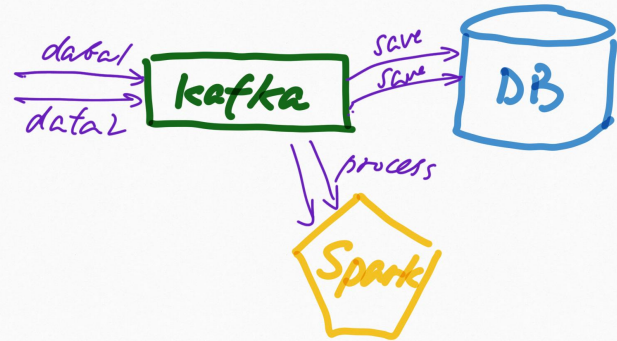


Spark Streaming



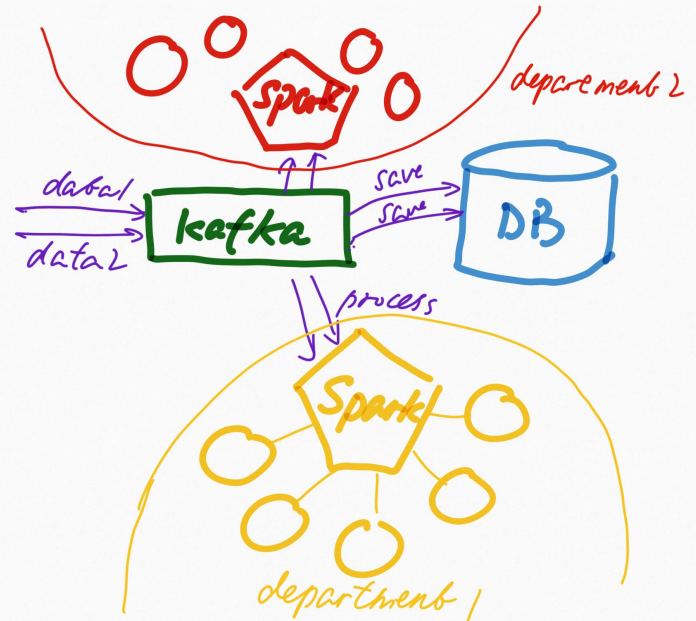
Where to Send Processed Data?

- Data is processed for consumption
 - Build Dashboard
 - Use as data model



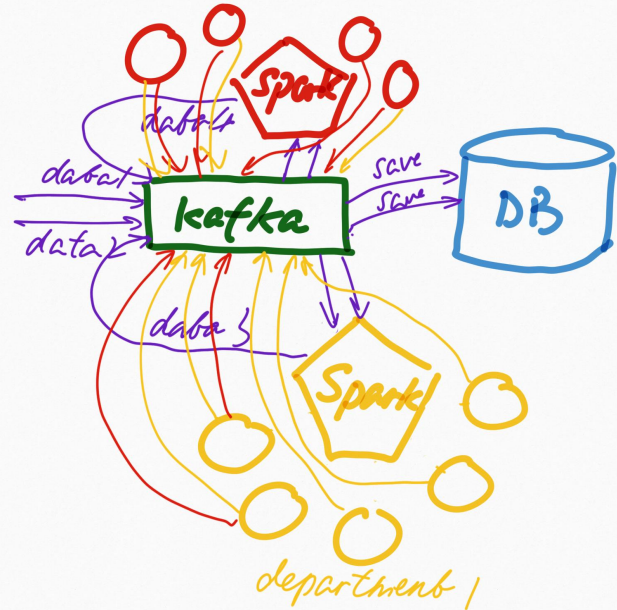
Where to Send Processed Data?

- Quickly create boundaries among teams
 - Data migration
 - Waste of resources



Send Data Back to Kafka

- No need to re-compute
- Encourage collaboration





Agenda

- Dev Environment
- Work with Spark
- **Work with Redis**
- Work with Node.js
- Interview Tips

Functionality

- Read data from Kafka
 - Should be able to read from any kafka cluster
 - Should be able to read from any kafka topic
- Publish to a Redis Pub
 - Should be able to write to any Redis Server
 - Should be able to write to any Redis Pub



Work with Redis Using Python

- `pip install virtualenv`
- `virtualenv env`
- `pip install redis`
- `pip freeze > requirements.txt`





Agenda

- Dev Environment
- Work with Spark
- Work with Redis
- **Work with Node.js**
- Interview Tips

Functionality

- Read data from Redis Sub
 - Should be able to read from any Redis Server
 - Should be able to read from any Redis Sub
- Update front-end UI as data come in
 - Socket.io
- Visualize data
 - smoothie.js, D3.js, Chart.js, Chartist.js



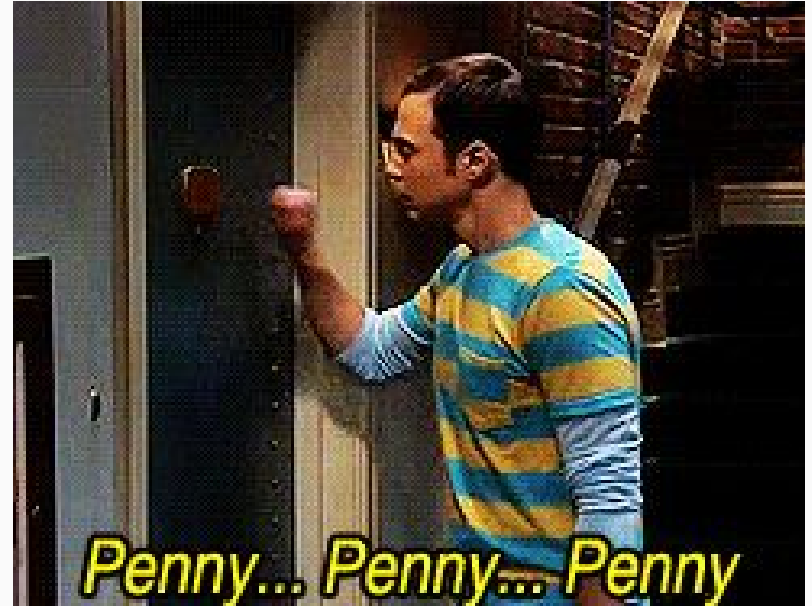
Work with Node.js

- `node -v`
- `npm -v`
- `npm install socket.io --save`
- `npm install express --save`
- `npm install minimist --save`
- `npm install smoothie --save`



How to Get Real Time Update

- Poll at an interval



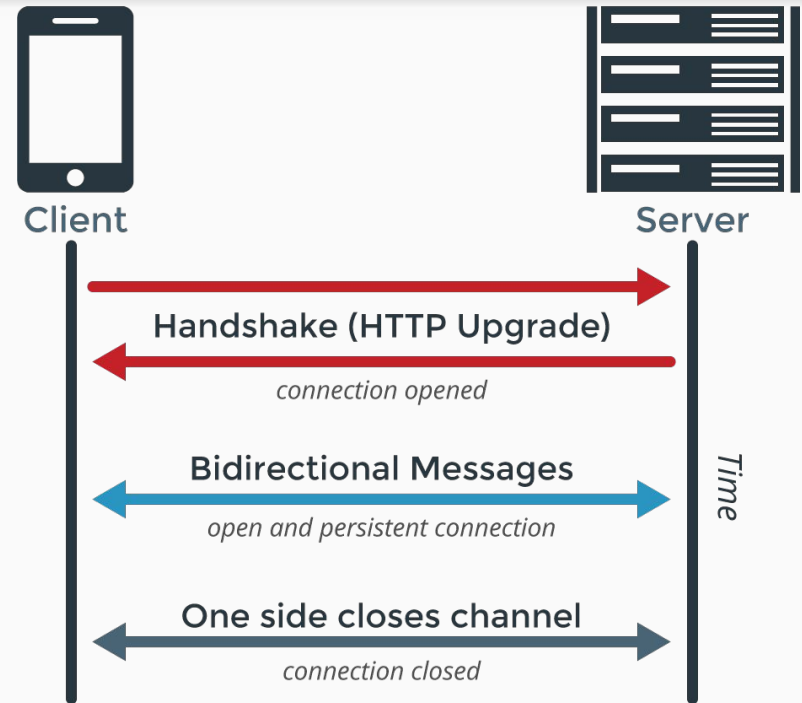
HTTP Requests and Response

- One request, one response



Websocket

- A long connection is established after initial handshake
- Server can 'push' data to client



Data Encoding and Schema

- We have been transferring JSON all over the place
 - Good cross-language parsing
 - Inefficient network IO
 - Other team cannot easily leverage your work
- Avro, Protocol Buffer, and Thrift to the rescue

Data Encoding and Schema

```
{  
  "userName": "Martin",  
  "favouriteNumber": 1337,  
  "interests": ["daydreaming", "hacking"]  
}
```

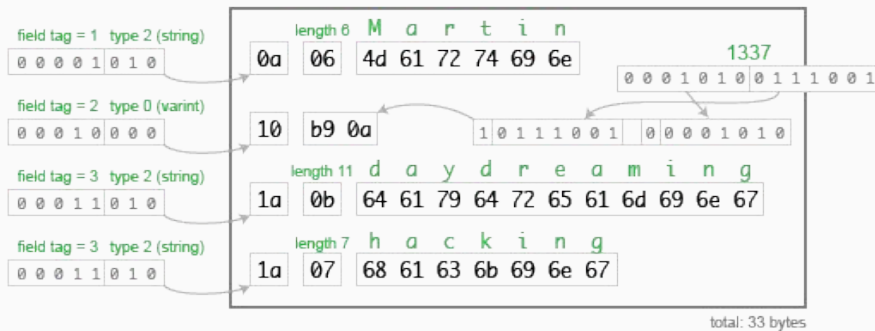
102 bytes

82 bytes without space and enter

Data Encoding and Schema

```
message Person {  
    required string user_name      = 1;  
    optional int64  favourite_number = 2;  
    repeated string interests      = 3;  
}
```

Protocol Buffers



Further Reading

- Kafka Connect:
 - <http://www.confluent.io/blog/announcing-kafka-connect-building-large-scale-low-latency-data-pipelines>
- Redis Common Web Uses
 - <http://highscalability.com/blog/2011/7/6/11-common-web-use-cases-solved-in-redis.html>
- Apache Avro: <https://avro.apache.org/>
- Apache Thrift: <https://thrift.apache.org/>
- Google Protocol Buffer:
<https://developers.google.com/protocol-buffers/>





Agenda

- Dev Environment
- Work with Spark
- Work with Redis
- Work with Node.js
- **Interview Tips**



Interview Tips

- Know Your Data
- Use Numbers
- Name Drop