Lab 5-Distributed Transaction Settlement System
Handout: May 31, 2018
Deadline: July 1, 2018 (No extension)

Assignment Overview:

This experiment requires you to use a variety of the latest distributed computing technologies to build and implement a comprehensive system on cloud server. The technologies that must be used include Spark Streaming, Kafka, MongoDB, and Zookeeper. Of course, you can also apply more other technologies to make certain features of the system be improved, and you will get Bonus.

Application scenario:

The SJTU market will generate a large number of foreign trade orders in real time. There are four countries (China, the United States, Japan, and Germany) involved in this market, and the currencies they used are RMB, USD, JPY and EUR. The market uses Union Currency (UNI) as the settlement currency. Each country's currency has an exchange rate with the Union currency, and the exchange rates change dynamically in every minute. Now you need to design and implement a distributed transaction settlement system, receive and process trade orders in real time, and calculate the transaction results of each country for each minute.

Regulations and examples:

(1) The initial time of the system is 2018-01-01 00:00:00, and the exchange rates at that time were as follows

Source Currency	Exchange Rate (to Union Currency)
RMB	2.0
USD	12.0
JPY	0.5
EUR	6.0

It means, for the transactions generated in the first minute, 1 RMB=2.0 UNI; 1 USD=12.0 UNI; 1 JPY=0.5 UNI; 1 EUR=6.0 UNI.

Requirements: **The exchange rates table should be managed by Zookeeper**. Please implement 4 parallel computing entities (multi-thread or multi-process) to modify the exchange rates of 4 countries. The rule is increasing the exchange rates 0.1 per minute.

(2) The calculation rule for each minute's transactions are showing as follows

Assume that 3 orders are generated within one minute from the start of 2018-01-01 00:00:00 (the above exchange rate table is valid):

Order 1:

Initiator: RMB
Recipient: USD
Turnover: 100（RMB） // The turnover is measured in the currency of the Initiator
Transaction Time: 2018-01-01 0:00:05

Order 2：

Initiator: USD

Recipient: JPY
Turnover: 200（USD）
Transaction Time: 2018-01-01 0:00:28

Order 3：
Initiator: RMB
Recipient: JPY
Turnover: 300（RMB）
Transaction Time: 2018-01-01 0:00:53

The results of this minute's transaction starting from 2018-01-01 0:00 are as follows:
{
China: income 0 RMB, expend 400 RMB // the result is in domestic currency
United States: income 16.67 USD, expend 200 USD
// The result shall be quoted in two decimal places
Japan: income 6000 JPY, expend 0 JPY // (200*12+300*2)/0.5 =6000
Germany: income 0 EUR, expend 0 EUR
}

（3） Input (trading order) and output

Order data will be given by HTTP request, and the data type is JSON string. The data format is defined as follows:

Order data format:
{
src_name: “CNY”,
dst_name: “USD”,
value: 100,
time: “2018-01-24 12:59:59”
}

The order data will continue to be sent. The order time will increase progressively from 2018-01-01 00:00:00. There may have orders with the exactly same transaction time.

Result data format:
{
name: “CNY”,
income: 16.67
expend: 200
time: “2018-01-24 12:59”
}

（4） Other requirements

- You need to set up a team, and each team contains no more than 3 people
- Deploy the system on JCloud (each team will have 3 cloud servers)
- Test data set has been given (3 JSON files), you need to write your own HTTP data sender (3 senders and each sender read one test data file) and data receiver (more

than 2, deployed on different server)

- Received data must be managed with **kafka**
- Data calculating platform must use **Spark Streaming**
- The result must be saved in **MongoDB**

Experimental report:

The experiment report should be divided into four parts.

(1) a document, describing the **system environment, install and configuration process**, **program design**, **the problems you encountered** and **the workload of each student**, etc.

(2) source code.

(3) Results calculated using the test data.

(4) Name and student ID of each team member

Upload your lab report as a gzipped tar file and name it as {Team leader's student ID}.tar.gz to:

ftp://liaoruohuai:public@public.sjtu.edu.cn/upload/lab5

[Grading]:

40%: the quality of lab report.

60%: laboratory examinations, scheduled on July 5th.