Lab 5-Distributed Transaction Settlement System

Handout: May 31, 2018

Deadline: July 1, 2018 (No extension)

Assignment Overview:

This experiment requires you to adopt a variety of the latest distributed computing support technology to build and implement a comprehensive system. The technologies that must be used include Spark Streaming, Kafka, MongoDB, and Zookeeper. Of course, you can also apply more technologies based on this 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) trade with each other in this market, and the currencies they use are RMB, USD, JPY and EUR. The market uses Union Currency (UNI) as the settlement currency, each country's currency has an exchange rate for the Union currency, and the exchange rate changes dynamically in minutes. Now you need to design and implement a distributed transaction settlement system, receive and process trade orders in real time, and calculate the transaction status of each country in each minute.

Regulations and examples:

(1) The initial time of the system is 2018-01-01 00:00:00, and the exchange rate table was as follows

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

It means: The transaction generated in this minute, 1 RMB=2.0 UNI; 1 USD=12.0 UNI; 1 JPY=0.5 UNI; 1 EUR=6.0 UNI.

Requirements: **The exchange rate table is managed with Zookeeper**. Implement four parallel computing entities (multi-thread or multi-progress) to modify the exchange rate of 4 countries. The modify rule is the exchange rate per minute plus 0.1

(2) The calculation rules for a minute's transaction are 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:
// the result is measured by domestic currency
China: income 0 RMB, expend 400 RMB
// the result accurate to two decimal places
United States: income 16.67 USD, expend 200 USD
// (200*12+300*2)/0.5 =6000
Japan: income 6000 JPY, expend 0 JPY
Germany: income 0 EUR, expend 0 EUR

（3） Input (trading order) and output

Order data is sent by HTTP Get 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 from 2018-01-01 00:00:00, there may be some orders that have exactly same time.

Result data format:

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

（4） Other requirements

- Test data have been given, you need to write your own order data sender and data receiver
- Received data must be managed with **kafka**
- Data calculation must use **Spark Streaming**

- The result must be saved to MongoDB

experimental report:

The experiment report is divided into three parts.

First, a document is used to describe: system environment and configuration process, Programming ideas and problems encountered, solutions, etc.

Second, source code.

Third, result of test data

Upload your lab report as a gzipped tar file and name it as {Your student ID}.tar.gz to ftp://solostagev9051:public@public.sjtu.edu.cn/upload/lab5

[Grading]:

40%: the quality of lab report.

60%: laboratory examinations, scheduled to early July.