【Experiment name】 SparkSQL analyzes GHTorrent data

【Purpose】

Master Docker to build a Spark stand-alone environment

Understand the GHTorrent project and related data structures

Mastering Spark Session configuration

Master the basic interface of SparkSQL

【Experimental content】

Topic : Write SparkSQL code to perform statistical analysis on developer data in GHTorrent.

Requirements : 1. Install Docker and build a stand-alone Spark environment;

2. Configure Spark Session;

3. Write Spark SQL code to process GHTorrent data, extract project feature data sets, such as project commit times, issue times, fork numbers, watch numbers, etc., and give a list of the top ten projects for each numerical feature from high to low;

4. Write Spark SQL code to process GHTorrent data, extract user feature data sets, such as user commit times, issue times, number of followers, number of fans, number of followed items, average fork number of followed items, etc., and give each numerical feature by A list of top ten users from highest to lowest.

Improvement (optional) : Associate the Segmentfault or Nuggets data crawled in Experiment 1, and count the relevant data of users with GitHub accounts on the website.

Experimental results (experimental steps and related core codes) :

Taking the statistics of "item commit times " and "user i issue times" in the user feature data set as an example, the code flow is divided into the following steps:

✓"Project commit times" code flow

define commit table

① Obtain table source path P ath

②Design table S tructType

③ Get the table (use cache () to reduce time-consuming)

```
[131]: commitsFilePath = "./phpData/commits_s3/part-00000-3453a56e-ce28-434a-a299-ee5bccca36bb-c000.csv"

commitsSchema = T.StructType([
    T.StructField("user_id", T.IntegerType(), True),
    T.StructField("item_id", T.IntegerType(), True),
    T.StructField("commit", T.IntegerType(), True),
])

commits_s3 = spark.read.csv(path=commitsFilePath, schema=commitsSchema, sep=",", nullValue="\\N").cache()
```

Create commit_s3 temporary data table

```
[138]: # 为spark.sql接口创建临时数据表
commits_s3.createOrReplaceTempView("commits_s3")
```

Extract item c ommit times

① Select the commit_s3 table, select the item id , the sum of commit , and perform group b y operation on the item id

② Arrange the sum of commit in descending order

```
[151]: # 项目commit次数
commit_info = spark.sql("SELECT item_id, sum(commit) AS commit_n FROM commits_s3 GROUP BY item_id ORDER BY commit_n DESC").limit(10)
#commit_top10= commit_info.orderby(commit_info.commit_n.DESC()).limit(10)
commitpath="./phpData/statistics/item/commit_top10"
#commit_top10.write.csv(path = commitpath, header=True, sep='', mode="overwrite")
commit_info.write.csv(path = commitpath, header=True, sep=',', mode="overwrite")
commit_info.show()
```

Get c ommit result

```
+-------+--------+
|item_id|commit_n|
+-------+--------+
|1360482|   72510|
| 524804|   71204|
|5482310|   65197|
|  14364|   47499|
|6965976|   37722|
|  15369|   32255|
|3296255|   29906|
|   1823|   29409|
|    634|   28706|
|  11450|   28618|
+-------+--------+
```

✓"User i issue times" code flow

define issue_table _

① Obtain table source path P ath

②Design table S tructType

③ Get the table (use cache () to reduce time-consuming)

```
[132]: issuesFilePath = "./phpData/issues_s3/part-00000-29484172-272b-46ec-976c-4d2c4534857f-c000.csv"

issuesSchema = T.StructType([
    T.StructField("user_id", T.IntegerType(), False),
    T.StructField("item_id", T.IntegerType(), False),
    T.StructField("issue", T.IntegerType(), False),
])

issues_s3 = spark.read.csv(issuesFilePath, schema=issuesSchema, sep=",", nullValue="\\N").cache()
```

Create i issue_s3 temporary data table

```
[138]: # 为spark.sql接口创建临时数据表
commits_s3.createOrReplaceTempView("commits_s3")
issues_s3.createOrReplaceTempView("issues_s3")
```

Extract the number of user i issue

```
[144]: # 用户issue次数
issue_info = spark.sql("SELECT user_id, sum(issue) AS issue_n FROM issues_s3 GROUP BY user_id ORDER BY issue_n DESC").limit(10)
issuepath="./phpData/statistics/user/issue_top10"
issue_info.write.csv(path = issuepath, header=True, sep=',', mode="overwrite")
issue_info.show()
```

get i issue result

```
+-------+-------+
|user_id|issue_n|
+-------+-------+
| 728107|   7013|
|5111376|   5604|
|  42735|   2522|
|4370493|   2429|
|  37983|   2336|
|    335|   1963|
| 408311|   1959|
|4439596|   1934|
| 338988|   1863|
|1960789|   1731|
+-------+-------+
```

Screenshot of statistical results:

Item c ommit times	Project i issue times	Project f ork number

item_id	commit_n
1360482	72510
524804	71204
5482310	65197
14364	47499
6965976	37722
15369	32255
3296255	29906
1823	29409
634	28706
11450	28618

Items are watched

item_id	issue_n
634	14523
524804	11754
1992097	9342
20096	8316
5710526	7009
2184	6325
23203	5801
15762	5655
11770480	5604
11450	5311

User c ommit times

item_id_source	forkers_n
14984418	251
16782015	23
4516	21
6845	20
2184	17
1691	16
20263549	14
105521384	13
900	11
46971728	10

User i issue times

item_id	user_n
4516	9522
6806	7084
634	6906
56050	6060
12695	5176
2696	5156
477175	4962
1992097	4824
18104	4818
12343173	4289

User attention

user_id	commit_n
49027	36856
28974	33849
1586	33457
5330	29836
4828	28431
1442006	25826
152222	23802
353479	22561
7249	20886
61097	20325

Number of fans

user_id	issue_n
728107	7013
5111376	5604
42735	2522
4370493	2429
37983	2336
335	1963
408311	1959
4439596	1934
338988	1863
1960789	1731

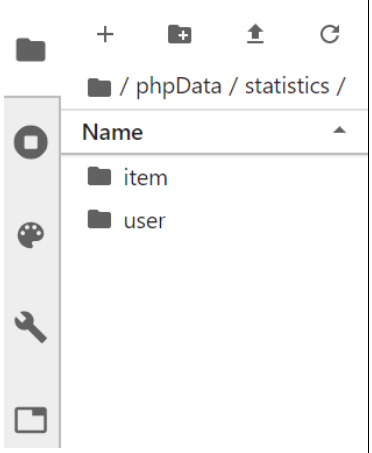
Number of items followed by users

user_id	user_n
1640412	14939
2313223	3020
8009234	2856
36006750	1764
11541601	1657
33817422	1614
268191	1607
10977842	1587
182107	1347
10561	1256

user_id_ed	user_n
28974	4785
4828	4385
9452	3471
1779	3331
6240	3124
3871	2808
2223	2316
2852307	1919
7020	1779
28798	1744

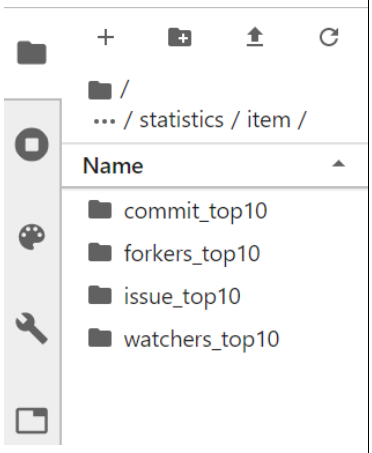
user_id	item_n
5483728	2017
9032	2002
81544	1834
260414	1721
76937	1534
1457687	1500
8869359	1352
7486482	1346
3962265	1326
4512290	1316

Statistical result storage



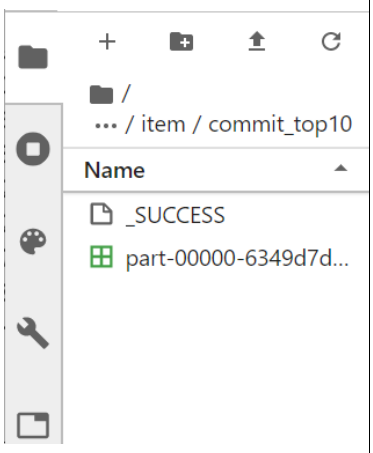
Statics folder

(level one directory)



Item folder

(secondary directory)



Commit_top10 folder

(Third-level directory)