به نام خدا

رایانش ابری

تمرین سوم

آشنایی با Hadoop و Spark (فاز صفر و یک)

طراحان تمرین:

محمدصادق محمدی، محمد رحمانیان

استاد درس:

دکتر جوادی

مهلت نهایی ارسال پاسخ:

۳۱ اردیبهشت ساعت ۲۳:۵۹

مقدمه:

در بخش اول این تمرین شما با **Hadoop** آشنا خواهید شد. Hadoop یک فریمورک نرم‌افزاری متن‌باز است که برای پردازش و ذخیره‌سازی داده‌های بزرگ و پیچیده مورد استفاده قرار می‌گیرد. این فریمورک از تکنولوژی‌های توزیع شده برای پردازش داده‌ها استفاده می‌کند، که به کمک آن می‌توان داده‌های ساختارمند و یا ناساختارمند را در سرورهای متعددی ذخیره و پردازش کرد. درست مانند سایر فریمورک‌های پردازش داده، هدف اصلی Hadoop نیز افزایش کارایی و قابلیت اطمینان در پردازش داده‌های بزرگ است. Hadoop از دو بخش اصلی تشکیل شده است:

- HDFS: این سیستم فایل توزیع شده برای ذخیره‌سازی داده‌های بزرگ با قابلیت اطمینان استفاده می‌شود. این فایل سیستم اطلاعات را به چندین نود در یک شبکه از نودها توزیع می‌کند. در این تمرین شما با این نوع سیستم فایل آشنا شده و با آن کار خواهید کرد.
- MapReduce: این یک مدل برنامه‌نویسی است که برای پردازش داده‌های بزرگ و موازی ساخته شده است. برنامه‌های MapReduce داده‌ها را به صورت موازی بر روی نودهای مختلف پردازش می‌کنند و نتایج را جمع‌آوری می‌کنند. در این تمرین قرار است که یک برنامه را با استفاده از این روش نوشته و آن را در کلاستر Hadoop خود اجرا کنید.

در بخش دوم شما با **Spark** آشنا خواهید شد. اسپارک یک فریمورک نرم‌افزاری متن‌باز برای پردازش داده‌های بزرگ است. اسپارک با استفاده از زبان‌هایی مانند Python, Java, Scala و R قابل استفاده است (در این تمرین ما از python استفاده خواهیم کرد). اسپارک به عنوان یک فریمورک متن‌باز و چند منظوره شناخته می‌شود که قابلیت پردازش داده‌های بزرگ را در سرعت بالا و با استفاده از محاسبات موازی فراهم می‌کند.
برخلاف Hadoop که از مدل MapReduce برای پردازش داده‌ها استفاده می‌کند، اسپارک از مدلی به نام RDD استفاده می‌کند. این مدل به برنامه‌نویسان اجازه می‌دهد تا با داده‌ها به صورت مشابهی عمل کنند و از پردازش موازی و یکپارچه‌سازی استفاده می‌کند تا عملکرد بهینه را فراهم کند. دلیل اصلی محبوبیت اسپارک در پردازش داده‌ها نسبت به ابزارهای دیگر مانند Hadoop این است که Hadoop در هر قسمت داده‌ها را در دیسک ذخیره کرده ولی اسپارک در RAM ذخیره می‌کند، همین قابلیت باعث سریعتر شدن اسپارک شده است. اسپارک به دلیل عملکرد بالا، قابلیت پردازش در زمان واقعی، پشتیبانی از مجموعه‌ای از الگوریتم‌ها و کتابخانه‌های مختلف (به عنوان مثال، برای یادگیری ماشین و تحلیل داده) و همچنین امکانات موازی سازی، بسیار محبوب است. اسپارک می‌تواند بر روی یک ماشین تنها، یک کلاستر کوچک یا حتی یک محیط ابری اجرا شود.

در فاز صفر تمرین سوم شما فقط باید برای هرکدام از Hadoop و اسپارک یک کلاستر بالا بیاورید تا در فازهای بعدی تمرین تسک‌های خواسته شده را انجام دهید.

شرح تمرین:

برای این تمرین یک فایل docker compose آماده شده است که برای شما پیش نیازهای زیرساختی تمرین را فراهم می‌کند. از جمله یک کلاستر Hadoop برای ذخیره‌سازی فایل‌ها در فضای ابری و به صورت توزیع شده که از چندین کانتینر تشکیل شده‌اند. و دیگری یک کلاستر اسپارک برای پردازش فایل‌ها به صورت موازی و توزیع شده از آن استفاده می‌شود. در آخر نیز یک jupyter notebook برای راحتی کار شما قرار گرفته است که به طور مستقیم کد خود را اجرا کنید.

بخش اول(اجرای کلاستر):

برای بالا آوردن کلاسترها قدم‌های زیر را انجام دهید:

1. ابتدا دستور زیر را داخل ترمینال خود اجرا کنید:

```
git clone git@github.com:sadegh-msm/hind.git
```

در صورت استفاده از معماری Arm64 دستور زیر را اجرا کنید:

```
git clone -b arm64 git@github.com:sadegh-msm/hind.git
```

2. در قسمت بعد فایل master-build.sh را با دستور زیر اجرا کنید:

```
bash master-build.sh
```

3. در صورت نیاز به پایین آوردن کلاستر نیز می‌توانید از دستور زیر استفاده کنید:

```
bash master-delete.sh
```

بعد از انجام قدم‌های بالا و می‌توانید با استفاده از URLهای زیر به واسط کاربری کانتینرهای بالا آمده دست بیابید.

Hadoop UI on ' http://localhost:9870 '
Spark Master UI on ' http://localhost:8080 '
Jupyter UI on ' http://localhost:8888 '

گزارش فاز صفر:

موارد زیر را باید در گزارش مربوط به این فاز بنویسید:

- نمایش کانتینرهای ایجاد شده
- توضیح وظایف هر کدام از کانتینرهای ایجاد شده
- نمایش UI برای Hadoop و Spark و Jupyter
    - توضیحات مربوط به تعداد نودهای اسپارک و منابع استفاده شده
    - توضیحات مربوط به اطلاعات NameNode و فایل سیستم آن

بخش دوم (اجرای MapReduce بر روی کلاستر):

در این تمرین، شما با استفاده از فرآیند MapReduce، اسنادی که هر کلمه در آن‌ها حداقل یک بار ظاهر شده است را محاسبه خواهید کرد. برای این کار باید اسکریپت‌های mapper و reducer خود به زبان پایتون را بر روی کلاستر هدوپ لوکال خود deploy کنید. این فرآیند به شما کمک می‌کند تا درک بهتری از توزیع کلمات در یک مجموعه داده پیدا کنید و برای تحلیل‌های متنی بسیار مفید است. در بخش زیر مراحل انجام این کار توضیح داده شده است:

۱. تکمیل اسکریپت‌های Mapper و Reducer به زبان پایتون:

در ابتدا فایل‌های mapper.py و reducer.py را تکمیل کنید. برای کامل کردن این قدم کارهای زیر را انجام دهید:

a. نوشتن اسکریپت Mapper:

ورودی: شناسه سند و متن سند.

فرآیند: برای هر کلمه در متن، یک جفت منحصر به فرد از کلمه و شناسه سند تولید کند.

b. نوشتن اسکریپت Reducer:

ورودی: کلمه و لیستی از شناسه‌های اسناد.

فرآیند: برای هر کلمه، شناسه‌های منحصر به فرد اسناد را محاسبه کنید و این اسناد را به همراه کلمه خروجی دهید.

۲. انتقال اسکریپت‌ها به NameNode:

سپس اسکریپت‌های mapper.py و reducer.py را در NameNode قرار دهید.

۳. آماده‌سازی داده‌ها در HDFS:

پس از آن داده‌های ورودی که در فایل input.txt قرار گرفته را در HDFS قرار دهید. برای این کار از دستورات hdfs استفاده کنید.

۴. اجرای Job MapReduce با استفاده از Hadoop Streaming:

یک Job MapReduce با استفاده از Hadoop Streaming ایجاد کرده و اسکریپت‌ها را اجرا کنید. برای این کار از دستور زیر استفاده کنید.

```
hadoop jar
/opt/hadoop-3.3.6/share/hadoop/tools/lib/hadoop-streaming-3.3.6.ja
r   -file mapper.py -mapper mapper.py   -file reducer.py -reducer
reducer.py   -input input/input.txt -output output
```

نکته: چون Hadoop بر روی سرور آپاچی که در جاوا ساخته شده است اجرا می شود، برنامه یک فایل Java JAR را به عنوان ورودی می گیرد. برای اجرای Python در Hadoop، باید از کتابخانه Hadoop Streaming

برای انتقال فایل اجرایی پایتون در چارچوب جاوا استفاده کنیم. در نتیجه، باید ورودی پایتون را از STDIN پردازش کنیم.

۵. بررسی خروجی:

پس از اتمام کار، خروجی تولید شده در output را مشاهده کنید.

راهنمایی:

خواندن داده‌ها: داده‌ها از طریق sys.stdin خوانده می‌شوند، که این روش استاندارد برای خواندن داده‌های ورودی در اسکریپت‌هایی است که با Hadoop Streaming استفاده می‌شوند.

نوشتن خروجی: خروجی‌ها باید به sys.stdout نوشته شوند، که در پایتون با استفاده از تابع print انجام می‌شود. این تضمین می‌کند که خروجی‌ها به درستی توسط فرآیند Reducer دریافت و پردازش شوند.

نمونه‌ای از ورودی:

```
doc1,The apple is sweet like the banana but unlike the bitter orange
doc2,Both the banana and the mango share a sweet taste unlike the sour orange
doc3,The apple and the orange are often compared for their vitamins
doc4,Mango and orange are tropical fruits unlike the temperate apple
doc5,Unlike the subtle mango the banana is often more overtly sweet
```

نمونه‌ای از خروجی (ترتیب اهمیت ندارد):

```
apple   {'doc1', 'doc3', 'doc4'}
banana  {'doc1', 'doc2', 'doc5'}
orange  {'doc1', 'doc2', 'doc3', 'doc4'}
mango   {'doc2', 'doc4', 'doc5'}
sweet   {'doc1', 'doc2', 'doc5'}
unlike  {'doc1', 'doc2', 'doc4', 'doc5'}
the {'doc1', 'doc2', 'doc3', 'doc4', 'doc5'}
is  {'doc1', 'doc5'}
like    {'doc1'}
but {'doc1'}
bitter  {'doc1'}
both    {'doc2'}
and {'doc2', 'doc3', 'doc4'}
share   {'doc2'}
a   {'doc2'}
taste   {'doc2'}
sour    {'doc2'}
are {'doc3', 'doc4'}
often   {'doc3'}
compared    {'doc3'}
for {'doc3'}
their   {'doc3'}
vitamins    {'doc3'}
tropical    {'doc4'}
fruits  {'doc4'}
temperate   {'doc4'}
subtle  {'doc5'}
more    {'doc5'}
overtly {'doc5'}
```

بخش امتیازی:

در این بخش تمرین، شما با استفاده از فرآیند MapReduce باید برترین 'K' کلمه پرتکرار را در هر سند را شناسایی کنید.

مانند بخش قبل اسکریپت‌های mapper و reducer را تکمیل کنید و سپس فایل input2.txt را با استفاده از آن‌ها اجرا کنید.

نمونه ورودی:

```
doc1,apple apple apple banana banana orange mango mango mango mango mango
doc2,technology technology technology technology data data data science machine machine learning
doc3,pandemic pandemic pandemic remote remote remote remote work work work work work workplace
doc4,climate climate climate climate change change change environment environment environment research research
doc5,robotics robotics robotics robotics robotics artificial intelligence intelligence intelligence solutions solutions
```

نمونه خروجی برای K=3 (ترتیب خروجی اهمیتی ندارد):

```
doc1,mango,5
doc1,apple,3
doc1,banana,2
doc2,technology,4
doc2,data,3
doc2,science,1
doc3,work,5
doc3,remote,4
doc3,pandemic,3
doc4,climate,4
doc4,change,3
doc4,environment,3
doc5,robotics,5
doc5,intelligence,3
doc5,solutions,2
```

گزارش فاز یک:

موارد زیر را باید در گزارش مربوط به این فاز بنویسید:

- توضیح کد mapper و reducer هر بخش
- نمایش و توضیح خروجی تولید شده توسط MapReduce هر بخش

نکات مربوط به تمرین تحویلی:

- تمرین شما تحویل اسکایپی خواهد داشت؛ بنابراین از استفاده از کدهای یکدیگر یا کدهای موجود در وب که قادر به توضیح داده عملکرد آنها نیستید، بپرهیزید.
- در صورت داشتن هرگونه مشکل، سوالی یا ابهام، آن را در با تدریس یاران درس مطرح کنید تا آن‌ها در سریع‌ترین زمان ممکن به شما پاسخ دهند.

مواردی که باید ارسال شود:

- یک فایل زیپ با نام studentID_HW3_1.zip که شامل گزارش و کدهای شماست.

موفق باشید

تیم تدریس‌یاری مبانی رایانش ابری