



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

درس رایانش ابری

تمرین سوم آشنایی با map reduce و اسپارک

طراحی تمرین
خانم هاشمی پور و آقای اسد

استاد درس
آقای دکتر جوادی

پاییز ۱۴۰۲

نکات تمرین

- سوالات خود را می‌توانید در گروه مشترک با تدریس‌یاران مطرح کنید.
- پاسخ تمرین را در قالب یک فایل زیپ با فرمت zip.3SID_HW ارسال بفرمایید.
- لطفاً پاسخ خود را تا تاریخ **۸ دی / ۱۴۰۲** در سامانه بارگذاری فرمایید.
- پاسخ به تمرین‌ها انفرادی است. از استفاده از کدهایی که بر آن‌ها مسلط نیستید بپرهیزید.
- تمرین تحویل آنلاین خواهد داشت.

بخش اول، آشنایی با MapReduce

در این بخش به پیاده‌سازی دو الگوریتم گرافی با استفاده از MapReduce می‌پردازیم.

مسئله کوتاه‌ترین مسیر

حتما با مسئله کلاسیک کوتاه‌ترین مسیر آشنایی کامل دارید. در این مسئله بین راس‌های گراف با طی یال‌های مختلف می‌توان جابجا شد و از مبدا به مقصد رسید. الگوریتم Dijkstra نمونه‌ای برای حل این مورد است. در این بخش، با استفاده از این الگوریتم در دو فاز map و reduce بر روی مجموعه داده‌ای از نقاط مختلف یک کشور، مسیریابی کرده و بهترین آن‌ها را انتخاب می‌کنیم. تفاوتی که پیاده‌سازی این الگوریتم با آنچه در درس ساختمان داده و الگوریتم فراگرفته‌اید، اجرای آن به صورت جاب‌های موازی MapReduce است. خوب است در ابتدا مروری بر پیاده‌سازی الگوریتم کلاسیک Dijkstra داشته باشید. در ادامه سودوکد این الگوریتم آورده شده است.

```
1: DIJKSTRA( $G, w, s$ )
2:    $d[s] \leftarrow 0$ 
3:   for all vertex  $v \in V$  do
4:      $d[v] \leftarrow \infty$ 
5:    $Q \leftarrow \{V\}$ 
6:   while  $Q \neq \emptyset$  do
7:      $u \leftarrow \text{EXTRACTMIN}(Q)$ 
8:     for all vertex  $v \in u.\text{ADJACENCYLIST}$  do
9:       if  $d[v] > d[u] + w(u, v)$  then
10:         $d[v] \leftarrow d[u] + w(u, v)$ 
```

همانطور که در بالا مشاهده می‌شود، با استفاده از یک صف دارای اولویت که در هر لحظه یک لیست مرتب‌شده از فاصله راس‌ها نگه می‌دارد، کوتاه‌ترین مسیر از نقطه شروع تا پایان انتخاب می‌شود. این عملکرد در MapReduce قابل پیاده‌سازی نیست چون نیازمند دانش کلی از همه نودها است. چیزی که مانع پیاده‌سازی و اجرا به صورت موازی می‌شود.

کدی برای الگوریتم Dijkstra به صورت موازی و در قالب دو فاز map و reduce بر روی گراف جهت‌دار ارائه دهید و آن را شرح دهید.

راهنمایی: شما می‌توانید از روش‌های مختلفی به این مسئله نگاه کنید. یک راه حل ساده این است که در ابتدا فرض کنید تمامی یال‌ها وزن برابر یک دارند. در این حالت راس‌های متصل به راس ابتدایی فاصله یک دارند. راس‌های متصل به این راس‌ها نیز فاصله ۱ + ۱ دارند و به همین صورت تا راس مقصد. اما چیزی که باید توجه داشته باشید آن است که در این حالت هم ممکن است بین دو راس، چندین مسیر با تعداد راس میانی مختلف قرار بگیرند که موثر در طول مسیر پیشنهادی است. پس باید از میان آنها، کوتاه‌ترین انتخاب شود. بدین صورت فازهای map و reduce را طراحی کنید و در گام بعدی تلاش کنید این فرض را از میان برده و برای حالت کلی با وزن‌های مختلف آن را پیاده‌سازی کنید.

مسئله رتبه صفحه (PageRank)

احتمالا نام این الگوریتم نیز به گوشتان خورده است. الگوریتمی پیشنهادی توسط گوگل که به وسیله ارجاع‌هایی که در هر صفحه وبسایت قرار دارد، صفحات وب را مرتب می‌کند. فرض کنید یک کاوشگر وب یک صفحه را می‌بیند و به صورت تصادفی بر روی یک لینک از آن صفحه کلیک می‌کند و این کار را بینهایت بار تکرار می‌کند. این الگوریتم با یک توزیع احتمالاتی روی مجموعه رئوس (در اینجا صفحه وب) بیان می‌کند که یک صفحه به چه احتمالی توسط کاوشگران بازدید خواهد شد. ایده اصلی این الگوریتم این است که اگر صفحه‌ای کیفیت خوبی داشته باشد، احتمالا صفحات دیگری با رتبه بالا این صفحه را تایید کرده اند. به همین صورت و طی یک الگوریتم پیاپی، امتیاز هر راس براساس امتیاز رئوس متصل بدست می‌آید. امتیاز هر راس به صورت زیر محاسبه می‌شود.

$$P(n) = \alpha \left(\frac{1}{|G|} \right) + (1 - \alpha) \sum_{m \in L(n)} \frac{P(m)}{C(m)}$$

که در آن $|G|$ تعداد کل گره‌ها (صفحات) در گراف، α ضریب پرش تصادفی، $L(n)$ مجموعه صفحاتی است که به n لینک دارند، و $C(m)$ تعداد لینک‌های صفحه m است.

در ابتدای هر iteration، یک راس مقدار امتیاز PageRank خود به سایر همسایه‌های متصل خواهد داد و در انتهای iteration، مقدار تمامی این امتیازهای دریافت شده هر راس، امتیاز جدید آن را تعیین می‌کنند. این الگوریتم تا جایی ادامه پیدا می‌کند که تغییر چندانی در امتیاز هر یک دیده نشود.

با استفاده از توضیحات بالا، در این بخش این الگوریتم را برای اجرا به صورت موازی و در قالب MapReduce بر روی یک گراف جهت‌دار پیاده‌سازی کنید.

نکات

- کد شما به صورت زیر ورودی بگیرد. به ازای هر ورودی، خروجی هر مرحله map و reduce باید قابل مشاهده باشد. مثال‌های زیر بدون هیچ مشکلی باید در ورودی قرار گرفته و اجرا شوند.

• مسئله کوتاه‌ترین مسیر

```
# Example graph
graph = {
    "A": {"Distance": 0, "AdjacencyList": {"B":10,"C":5}},
    "B": {"Distance": float('inf'), "AdjacencyList": {"C":2,"D":1}},
    "C": {"Distance": float('inf'), "AdjacencyList": {"E":2,"B":3,"D":9}},
    "D": {"Distance": float('inf'), "AdjacencyList": {"E":4}},
    "E": {"Distance": float('inf'), "AdjacencyList": {"A":7,"D":6}}
}
```

- مسئله رتبه صفحه

```
# Example graph
graph = {
  "A": {"PageRank": 1.0, "AdjacencyList": ["B", "C"]},
  "B": {"PageRank": 0.5, "AdjacencyList": ["A"]},
  "C": {"PageRank": 0.5, "AdjacencyList": []}
}
```

- اجرای الگوریتم بر روی کلاستر هدوپ نمره امتیازی دارد. بدین منظور docker compose قرار داده شده تمامی منابعی که نیاز دارید را بالا خواهد آورد. دیتاست‌های مجزایی نیز قرار در پوشه hadoop/data قرار گرفته‌اند که جاب‌های MapReduce بر روی آنها اجرا خواهند شد.
- می‌توانید از یکی از زبان‌های جاوا و یا پایتون برای پیاده‌سازی استفاده کنید.
- در گزارشی که تهیه می‌کنید باید به توضیح الگوریتم پیشنهادی بپردازید.
- کد map reduce را حتما آپلود کنید.

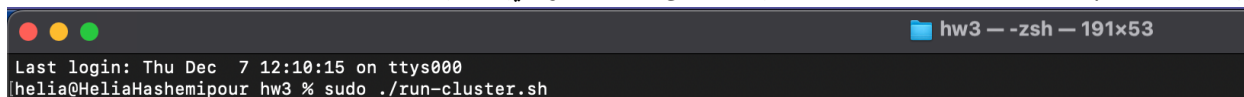
بخش دوم، آشنایی با اسپارک

آپاچی اسپارک یک فریمورک متن باز است که در سال 2009 توانست با بهینه‌سازی 100 برابر نسبت به پلتفرم روز آن زمان یعنی Hadoop جایگاه بالایی در میان علاقه‌مندان به مباحث BigData باز کند. دلیل قدرت این فریمورک وجود هسته مرکزی با یک موتور پردازش توزیع یافته کلان داده است که می‌تواند به خوبی مقیاس‌بندی شود. همچنین یکی دیگر از مواردی که باعث افزایش کارایی اسپارک شده استفاده از مکانیسم حافظه نهان هنگام استفاده از داده‌هایی است که قرار است دوباره در برنامه استفاده شود. این کار باعث کاهش سرشار ناشی از خواندن و نوشتن از دیسک می‌شود. با استفاده از مکانیسم حافظه نهان در اسپارک داده‌ها یک بار از دیسک خوانده می‌شوند و در حافظه اصلی کش می‌شوند و عملیات‌های متفاوت بر روی آن اجرا می‌شود. در نتیجه استفاده از این روش نیز باعث کاهش چشمگیر سرشار ناشی از ارتباط با دیسک در برنامه‌ها و بهبود کارایی می‌شود. با توجه به قدرت این پلتفرم یکی از پتانسیل‌های استفاده از آن مبحث استریم دیتا می‌باشد. امروزه شرکت‌های تکنولوژی زیادی از آن با مقاصد تجاری استفاده می‌کنند. از طرف دیگر ویژگی ساده بودن یادگیری آن موجب شده تا این فریمورک به عنوان یک مهارت میبایست در رزومه توسعه‌دهندگان قرار داشته باشد. در این تمرین تلاش شده تا از توانایی‌های ابتدایی این پلتفرم استفاده شود.

برای این تمرین یک فایل داکر کامپوز آماده شده است که برای شما پیشنیازهای زیرساختی تمرین را فراهم می‌کند. از جمله 1 کلاستر hdfs برای ذخیره‌سازی فایل در فضای ابری به صورت توزیع شده که از یک کاننتینر namenode و دو کاننتینر datanode تشکیل شده است. 2 کلاستر اسپارک که از یک مستر و دو ورکر تشکیل شده است که برای پردازش فایلها به صورت موازی و توزیع شده از آن استفاده می‌شود. یک فایل جویپتر نوتبوک که در نتورک داکری همه این‌ها قرار گرفته تا به راحتی متصل باشد و این فایل باید توسط شما به زبان پایتون کامل گردد.

```
- Spark Master: [http://localhost:9090] (http://localhost:9090)
- Namenode: [http://localhost:9870] (http://localhost:9870)
- Jupyter Notebook: [http://localhost:8888] (http://localhost:8888)
```

برای راه اندازی Hadoop و بالا آوردن کاننتینرهای ذکر شده، کافی است دستور زیر را اجرا کنید:



```
hw3 — -zsh — 191x53
Last login: Thu Dec 7 12:10:15 on ttys000
helia@HeliaHashemipour hw3 % sudo ./run-cluster.sh
```

در صورتی که سیستم شما اینتل نبود این خط از داکرفایل را به این شکل عوض کنید.

/ENV JAVA_HOME=/usr/lib/jvm/java-8-openjdk-arm64

توضیحات دیتاست‌ها

برای این تمرین دو دیتاست در نظر گرفته شده است.

دیتاست اول (مرگ و میر ویروس کرونا)

- این دیتاست شامل رکوردهایی با مضمون تعداد مرگ و میر ویروس کرونا میباشد.
- رکوردهای این دیتاست شامل ۱۲ ستون است.
- در برخی از رکوردهای دیتاست، ممکن است اطلاعات یک ستون وجود نداشته باشد.

دیتاست دوم (پترن‌های باکتری Ecoli)

برای آشنایی بهتر با قابلیت‌های PySpark این دیتاست به صورت فایل txt در نظر گرفته شده است. فایل Ecoli.txt به طور معمول داده‌های دنباله ژنومی باکتری اشریشیاکلی (E. coli) را شامل می‌شود. اشریشیاکلی یک باکتری است که به طور متداول در تحقیقات بیولوژیکی مورد مطالعه قرار می‌گیرد و دنباله ژنومی آن به صورت فایل‌های متنی مانند Ecoli.txt در دسترس است. فایل‌های داده‌های دنباله ژنومی مانند Ecoli.txt شامل یک رشته از نویسه‌ها هستند که بازگوکننده بازهای نوکلئوتیدی است که DNA سازنده این ارگانیسم است. در مورد اشریشیاکلی، این فایل شامل یک دنباله از بازهای آدنین (A)، سیتوزین (C)، گوانین (G) و تیمین (T) است که کد ژنتیکی آن را تشکیل می‌دهد. پژوهشگران از این فایل‌های دنباله ژنومی برای اهداف مختلفی استفاده می‌کنند، از جمله تحلیل ژنتیکی، شناسایی ژن‌ها، مطالعه روابط تکاملی و درک ترکیب ژنتیکی و عملکرد ارگانیسم‌ها مانند اشریشیاکلی.

موارد زیر را در فایل گزارش نمایش دهید:

- نمایش کانتینرهای ایجاد شده
- توضیح وظیفه هرکدام از کانتینرها در Hadoop
- نمایش WebUI برای NameNode و فایل سیستم آن

نکات

دیتای اولیه در دو فایل فشرده شده به صورت csv و txt در اختیار شما قرار داده شده است که دیتاست اول شامل اطلاعات مربوط به مرگ و میر ویروس کرونا و دیتاست دوم شامل پترن‌های باکتری Ecoli می‌باشد. در این تمرین از شما خواسته شده است، ابتدا کانتینرهای مورد نیاز را بالا آورده سپس در آدرس <http://localhost:8888> به سوالاتی که در فایل jupyter notebook مربوط به هر دو دیتاست می‌باشد، پاسخ دهید.

- انتظار می‌رود گزارش مربوطه برای این بخش، شامل سوالات داخل دستورکار و نوتبوک باشد.

پیروز باشید.

تیم تدریسیاری درس رایانش ابری