نصب راه اندازی و اجرای PySpark

پروژه درس تحلیل‌ها و سیستم‌های داده های حجیم

تهیه کننده: سید حمید مهدوی

استاد: دکتر حسن نادری


بسم الله الرحمن الرحیم
MRTsoR

فهرست مطالب

فهرست تصاویر

فهرست جداول

فهرست نمودارها

۱: مقدمه

اسپارک ابزاری قدرتمند برای کار بر روی داده‌های حجیم است. این ابزار مدل MapReduse هدوپ رو گسترش داده است و توانسته به سرعت بالاتری از هدوپ دست یابد ولی این نکته قابل توجه است که اسپارک جایگزینی برای هدوپ نیست بلکه در زیست‌بوم آپاچی در کنار هدوپ قرار می گیرد.
این ابزار به صورت پیشفرض از زبان‌های اسکالا و پایتون و آر پشتیبانی می‌کند. Pyspark کتابخانه‌ای است که به واسطه آن می‌توان با استفاده از پایتون از اسپارک استفاده کرد. در این نوشتار به بررسی فرآیند نصب و راه اندازی و استفاده از pyspark پرداخته خواهد شد.
این مستند و سایر مستندات و کد های مرتبط در مسیر

```
1 https://github.com/sahama/bigdata_project
```

در دسترس خواهند بود.

RDD :۱.۱

در اسپارک مفومی وجود دارد به نام «مجموعه داده های توزیع انعطاف پذیر»[1] که ساختمان داده بنیادی اسپارک است. هر مجموعه داده در RRD به صورت منطقی تقسیم به قسمت‌هایی می‌شود که هر کدام از قسمت‌ها ممکن از در یکی از گره مورد پردازش قرار گیرد و اصولاً به شکل فقط خواندنی است.
در کل به دو صورت می‌توان از RDD استفاده کرد

۱. **parallelizing** که یک مجموعه داده برای داده‌های موجود بر روی راه انداز برنامه است.
۲. **referencing a dataset** اگر قرار باشد از **HDFS** یا **HBASE** استفاده شود این ارتباط از طریق **RDD** خواهد بود.

یکی از دلایل اصلی که اسپارک از هدوپ سریع‌تر است همین **RDD** است که دسترسی های زیادی که در **MR** وجود دارد را تغییر می دهد.

۲: نصب

برای نصب این ابزار از سیستم عامل debian 8 استفاده شده است و دستورات و فرآیند در این سیستم عامل تست شده است.
ممکن است در پیکربندی اسپارک به هدوپ نیاز باشد. هم ابزار اسپارک و هم ابزار هدوپ را می‌توانید از سایت‌های مربوط در سایت آپاچی دانلود کنید ولی نسخه اسپارک هماهنگ با نسخه هدوپ را دریافت کنید. در این تجربه از هدوپ نسخه ۲.۸ و اسپارک نسخه ۲.۱.۱ و پایتون نسخه ۳.۴ که در مخازن دبیان موجود است استفاده شده است.

۲.۱: نصب هدوپ

همان‌طور که گفته شد برای نصب اسپارک ابتدا باید به سراغ نصب هدوپ برویم. به این منظور این ابزار را در مسیر

```
2 /usr/local/hadoop/
```

استخراج[2] می‌کنیم و در فایل bashrc. این تنظیمات را انجام می‌دهیم

```
3 export HADOOP_PREFIX=/usr/local/hadoop
4 export PATH=$PATH:$HADOOP_PREFIX/bin
```

همچنین تنظیمات لازم برای اجرای اسپارک نیز به این فایل اضافه می گردد:

1 Resilient Distributed Datasets
2 extract

```
export SPARK_PREFIX=/usr/local/spark
export PATH=$PATH:$SPARK_PREFIX/bin
export PATH=$PATH:$SPARK_PREFIX/sbin
export PYSPARK_PYTHON=python3
```

بعد از این مرحله می‌توان به سراغ نصب ابزار spark رفت.

۲.۲: نصب اسپارک

برای نصب اسپارک فایل فشرده دانلود شده اسپارک را در مسیر زیر استخراج کرد.

```
/usr/local/spark/
```

به این ترتیب با توجه به اینکه مسیر فایل‌های اجرای هدوپ و اسپارت را در path قرار داده‌ایم می‌توان دستورات این دو ابزار را مستقیم و بدون اشاره به مسیر اجرا کرد.
توجه به این نکته ضروری به نظر می‌رسد که اسپارک به صورت پیشفرض از پایتون پیشفرض که نسخه ۲.۷ است استفاده می‌کند ولی ما در این سند قصد استفاده از پایتون نسخه ۳ را داریم به همین منظور خط شماره ۷ در تنظیمات اضافه شده است.

۲.۳: نصب به عنوان ماژول پایتون

در مرحله فعلی می‌توان از پای اسپارک و اسپارک استفاده کرد ولی این کتابخانه در پایتون سیستم عامل یا virtualenv قابل شناسایی نیست.
برای این منظور باید ابزار pyspark که همراه با بسته دانلود شده اسپارک است را در حالت develop بر روی مفسر پایتون مورد نظر نصب کنیم.
لازم به ذکر است نصب در حالت develop مهم است چرا که این کتابخانه از آدرس دهی محلی برای دسترسی به فایل‌های اسپارک استفاده کرده است و نصب در حالت develop پوشه جاری را به بسته های نصب شده اضافه می‌کند ولی نصب در حالت production پوشه مربوط به کتابخانه را به دایرکتوری کتابخانه‌های پایتون مورد نظر منتقل می‌کند و در نتیجه در صورتی که pyspark به صورت production نصب شود مسیر ها به هم می‌خورد.
فرض کنیم که قصد ساخت یک venv جدید برای کار با اسپارک را داریم. به این منظور در مسیردلخواه دستور زیر را وارد می کنیم:

```
python3 -m venv pyspark_env
```

به این ترتیب یک مفسر جدید که می‌توان گفت یک کپی از پایتون سیستم عامل شما است در مسیر خواسته شده ساخته می شود.
اکنون برای استفاده از این مفسر جدید باید آن را فعال کرده و با استفاده از دستور pip مربوط به پایتون مورد نظر کتابخانه‌ی pyspark را نصب کرد.

```
. <where your python interpriter live>/bin/activate
cd /usr/local/spark/python/
pip install -e .
```

اکنون می‌توان به صورت مستقیم pyspark را در پایتون انتخابی import و اجرا کرد.

۳: اجرا

با استناد به فرآیند نصب که در بخش قبل توضیح داده شد اکنون در سه حالت می‌توانیم برنامه‌های خود را تحت pyspark اجرا کنیم

۳.۱: مستقیماً با استفاده از کنسول pyspark

به این منظور می‌توان دستور pyspark را در خط فرمان اجرا کرد

```
14 # pyspark
```

علامت # نشان می‌دهد که این دستور با دسترسی root اجرا شده است.

که بعد از اجرای این دستور خروجی شبیه به این در کنسول ظاهر می‌شود (لاگ ها برای کم کردن حجم حذف شده اند)

```
15 Python 3.4.2 (default, Oct  8 2014, 10:45:20)
16 [GCC 4.9.1] on linux
17 Type "help", "copyright", "credits" or "license" for more information.
18 Setting default log level to "WARN".
19 To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
      setLogLevel(newLevel).
20 .....................................
21 Welcome to
22       ____              __
23      / __/__  ___ _____/ /__
24     _\ \/ _ \/ _ `/ __/  '_/
25    /__ / .__/\_,_/_/ /_/\_\   version 2.1.1
26       /_/
27
28 Using Python version 3.4.2 (default, Oct  8 2014 10:45:20)
29 SparkSession available as 'spark'.
30 >>>
```

با این کار کنسول پایتون در حالت interactive فعال شده و در ادامه می‌توانید دستورات پایتون و دستورات pyspark استفاده کنید.

۳.۲: استفاده از spark-submit

در هنگام استفاده از spark-submit می‌توانید فایل اسکریپت برنامه خود را به اسپارک برای اجرا بدهید. ابزار run_example نیز از همین دستور استفاده می‌کند.

به این منظور دستوری شبیه به این را در خط فرمان می‌دهیم

```
31 spark-submit pi.py 10
```

ورودی اول دستور spark-submit است

ورودی دوم اسکرپت پایتونی که قصد اجرای آن را داریم که در این مثال از یکی از sample های خود اسپارک به نام pi.py استفاده کرده‌ایم

ورودی سوم آرگومان ورودی مورد نیاز اسکریپت پایتون است.

با اجرای این دستور خروجی شبیه به این ظاهر می‌شود

```
32 Pi is roughly 3.220000
```

لازم به ذکر اینکه در این حالت باید مفسر پایتون مورد نظر فعال شده باشد.

نیازی به توضیح نیست که لاگ ها برای کم کردن حجم در این سند حذف شده اند.

۳.۳: اجرای مستقیم کد پایتون

به این منظور به طور مستقیم اسکریپت پایتون نوشته شده را با مفسر پایتون انتخابی اجرا می‌کنیم. به این منظور از دستوری شبیه به این استفاده می کنیم.

```
33 python pi.py 10
```

و خروجی شبیه به خروجی روش اجرای قبل را خواهیم داشت.