

به نام خدا
دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

سیستم‌های توزیع شده

موضوع: پروژه

استاد درس: دکتر کلباسی

مهلت تحویل: ۱۲ بهمن ۱۳۹۸

نیم‌سال اول ۹۹-۱۳۹۸

Contents

۳ ساختار سامانه
۳ وظیفه‌ی Master:
۳ وظیفه‌ی Worker:
۴ فرضیات سامانه
۴ نحوه‌ی مشخص شدن workerها به چه صورت است؟
۵ چگونه فایل داده تقسیم بندی می‌شود، اگر داده بسیار پر حجم باشد چه کار باید انجام داد؟
۵ امکان دسترسی بیش از یک کلاینت به این خوشه به شکل ممکن خواهد بود؟
۶ فایلها و به خصوص داده کجا ذخیره میشود؟
۶ زبان برنامه نویسی کد رایانش چه زبانی است؟
۷ کدهای اجرایی در Master و Worker چه خاصیتی باید داشته باشند؟
۷ نحوه‌ی اجرای پروژه:
۹ مراحل اجرای یک فرآیند به شرح تصویر:
۱۱ مثال‌هایی از پردازش خوشه‌ای که روی این سامانه اجرا شده است.
۱۱ مثال اول: Word Count
۱۱ مثال دوم: Character Count
۱۱ مثال سوم: شناسایی مظنونین
۱۲ مثال چهارم: پیدا کردن بیشینه کاربر همزمان اپراتورهای تلفن همراه

ساختار سامانه

در این پروژه که شامل دو ماژول **Master** و **Worker** است، بستری ایجاد کردیم تا با آن بتوان فرآیندهای محدودی از پردازش خوشه‌ای را انجام داد. این بستر تشکیل شده‌است از یک **Master Node** و چندین **Worker Node** که توسط کاربر سامانه تعیین می‌شود. گرهی **Master** می‌تواند در صورت نیاز وظیفه‌ی گرهی **worker** را هم انجام دهد، ولی دقت کنید که در یکی گره نمی‌توانیم بیشتر از یکی **Worker** داشته باشیم.

وظیفه‌ی **Master**:

گره‌ی **Master** وظیفه‌ی مدیریت پردازش در این خوشه را به عهده دارد، به این صورت که هنگامی که کاربر اطلاعات لازمه را برای **Master** ارسال می‌کند، این گره ابتدا فایل داده را به چندین فایل کوچکتر تقسیم می‌کند (نحوه‌ی تقسیم فایل‌ها در ادامه آمده است) و سپس هر فایل شکسته شده را به همراه **Code** مربوطه به یک **Worker** ارسال می‌کند. سپس بعد از جمع‌آوری تمامی نتایج از **Worker** ها، با ادغام این فایل‌های نتایج با یکدیگر و سپس اجرای فاز دوم پردازش به روی آن فایل نتایج نهایی را ایجاد می‌کند و در اختیار کاربر سامانه قرار می‌دهد.

وظیفه‌ی **Worker**:

سامانه از چندین **Worker** تشکیل شده است و از لحاظ تعداد **Worker** کاملاً توسعه‌پذیر است. هر **Worker** آماده است تا فرایندی را که **Master** به آن محول می‌کند را انجام دهد.

هنگامی که فایل داده‌ای به همراه Code مربوطه توسط Master به یک Worker ارسال می‌شود، گرهی Worker مشخص شده ابتدا از لیست گرهی های idle که در Master node وجود دارد حذف می‌شود، حال Worker در صورت نیاز Code را کامپایل کرده و سپس آن را بر روی داده‌ی مورد نظر اجرا می‌کند و پس از ایجاد فایل نتایج، این فایل را به گرهی Master ارسال کرده و پس از آن دوباره به لیست گره‌های idle در Master اضافه می‌شود.

فرضیات سامانه

نحوه‌ی مشخص شدن workerها به چه صورت است؟

برای اجرای Master Process باید فایل تنظیماتی شامل آدرس Master (آدرس گره‌ای است که Master روی آن اجرا می‌شود) و آدرس تمامی workerها ایجاد شود و در هنگام اجرای Master Process به آن داده شود. همچنین برای اجرای Worker Process باید فایل تنظیماتی شامل آدرس Master و آدرس Worker (آدرس گره‌ای که این Worker روی آن اجرا می‌شود) ایجاد شود و در هنگام اجرای Worker Process به آن تحویل شود. حال Master Process می‌تواند با خواندن این فایل تنظیمات از تعداد و آدرس Workerها با خبر شود. نمونه‌ای از تنظیمات مربوطه به Master و Worker در ادامه آمده است.

```
1
2 master = "192.168.186.11"
3 mine = "192.168.186.13"
```

Figure 1 تنظیمات مربوطه به Worker

```

1
2 master = "192.168.186.11"
3
4 worker {
5     1 = "192.168.186.11"
6     2 = "192.168.186.12"
7     3 = "192.168.186.13"
8 }
9

```

Figure ۲ تنظیمات مربوطه به Master

چگونه فایل داده تقسیم بندی می شود، اگر داده بسیار پر حجم باشد چه کار باید انجام داد؟

در این سامانه داده ها بر اساس حجم تقسیم بندی می شوند که این مقدار هنگام ارسال کد و فایل داده ای اصلی به Master پرسیده می شود. سپس فایل داده ای اصلی بر اساس به چندین فایل کوچک تر تقسیم می شود، نحوه ای انجام این تقسیم بندی به این گونه است که Master Node فایل داده ای اصلی را خط به خط می خواند و هر جا که حجم از مقدار مشخص شده بیشتر شد، آن بازه از فایل را تبدیل به یک فایل کوچکتر می کند، در نتیجه سامانه در مواجهه با داده های پر حجم به مشکل نمی خورد، اما ممکن است یک فایل پر حجمی تنها از یک خط تشکیل شده باشد که این فایل ممکن است سامانه را با مشکل مواجه سازد، برای برطرف کردن این مشکل می توان یک delimiter از کاربر دریافت کرد و فایل اصلی را بر حسب آن و حجم مشخص شده تقسیم بندی کرد، اما ما در این سامانه برای راحتی کار این قابلیت را اضافه نکردیم و فایل ها را بر اساس حجم و خط به خط تقسیم کردیم.

امکان دسترسی بیش از یک کلاینت به این خوشه به شکل ممکن خواهد بود؟

متأسفانه امکان دسترسی بیشتر از یک کلاینت به این خوشه ممکن نیست، در واقع بهتر بود که برای اجرای هر پردازش ای در این خوشه به ازای هر پردازش در یک Worker یک Container ایجاد شود تا بتوان مدیریت بهتری روی پردازش ها در هر Worker داشت و همچنین در این صورت می توانستیم از تاثیر گذاری پردازش ها روی یکدیگر جلوگیری کنیم و همچنین می توانستیم روی

منابعی که در اختیار هر Container قرار می‌گیرد هم نظارت کنیم، ولی ما در این سامانه برای راحتی کار این قابلیت را اضافه نکردیم.

فایلها و به خصوص داده کجا ذخیره میشود؟

فایل ارسالی به Master Node و باقی فایل‌های ایجاد شده در این سامانه به روی فایل سیستم گرهی مربوطه ذخیره شده و فایل‌های واسطه پس از انجام فرآیند حذف می‌گردند، همچنین فایل نهایی حاوی نتیجه‌ی اجرای پردازش در گرهی Master ذخیره می‌شود و لینک دانلودش نیز برای کاربر سامانه پس از انجام فرآیند ارسال می‌شود.

بهتر بود برای ذخیره سازی داده‌ها از یک فایل سیستم توزیع شده استفاده می‌کردیم تا بتوانیم فایل‌های حجیم را نیز نگهداری کنیم و همچنین جلوی از بین رفتن فایل‌ها را با عواملی همچون replication بگیریم، همانطور که Hadoop برای انجام فرآیندهای Map-Reduce فایل‌ها را از DFS خوانده و در آن ذخیره سازی می‌کند. همچنین باید این را در نظر داشت که در واقع ایده‌ی اصلی Map-Reduce این است که به جای این که فایل را به جایی که Code قرار دارد منتقل کنیم، باید Code را به جایی که فایل داده قرار دارد منتقل کنیم، اما در پروژه‌ی ما بیشتر هدف ایجاد یک بستری برای پردازش‌های خوشه‌ای و ساخت یک خوشه با قدرت پردازشی بالا است و همچنین داده‌ها هم به همراه Code در بین Node ها جابه‌جا می‌شوند لذا کمتر نیاز به یک DFS در این پروژه احساس می‌شود، با این حال اگر بتوان بستری ایجاد کرد که فایل‌ها در این پروژه به صورت توزیع شده ذخیره شوند، می‌توان قابلیت‌های پروژه را افزایش داد.

زبان برنامه نویسی کد رایش چه زبانی است؟

در این سامانه برای کد رایش دو زبان Python و CPP در نظر گرفته شده و کاربر سامانه می‌تواند زبان برنامه نویسی خود را از بین این دو گزینه انتخاب کند، همچنین امکان افزودن دیگر

زبان ها هم در این سامانه وجود دارد که جهت راحتی کار فعلا تنها این دو زبان برای سامانه انتخاب و پیاده سازی شده است.

کدهای اجرایی در Master و Worker چه خاصیتی باید داشته باشند؟

کدهایی که برای اجرا در مرحله پرداز اولیه و مرحله نهایی یا جمع بندی ارسال می شوند باید با استفاده از `argv` دو متغیر دریافت کنند که متغیر اول آدرس فایل ورودی برنامه و متغیر دوم آدرس فایلی است که خروجی برنامه در آن ذخیره خواهد شد، در نتیجه این کدها باید ورودی را از فایل مشخص شده در `argv` بخوانند و نتیجه را در فایل خروجی مشخص شده در `argv` یادداشت کنند. همچنین کد مرحله دوم باید خروجی تولیدی از کد مرحله اول را بتواند بخواند و تفسیر کند.

نحوه اجرای پروژه:

پروژه به زبان جاوا و برای سیستم عامل های Linux Base نوشته شده است برای راه اندازی آن باید روی سیستم JAVA و MAVEN داشته باشید، برای ایجاد فایل های اجرایی مربوط به گره Master و Worker داخل دایرکتوری اصلی پروژه به اسم MapReduce شوید و دستور `mvn clean install` را اجرا کنید تا فایل های اجرایی ساخته شود، حال می توانید فایل اجرایی Worker را از پوشه `MapReduce/worker/target/` با اسم `ff.jar` دریافت کنید، همچنین برای دریافت فایل اجرایی Master به پوشه `MapReduce/master/target/` مراجعه کرده، فایل اجرایی Master با اسم `L.jar` در این پوشه قرار دارد. (به دلیل حجم بالا این دو فایل اجرایی، این دو فایل را از قبل داخل zip آپلود شده قرار ندادیم)

حال برای اجرای Master Process به گرهی Master بروید و دستور زیر را اجرا کنید. (دقت کنید که باید آدرس فایل `Master.conf` را در هنگام اجرای Process تعیین کرد)

```
java -jar master-1.0-SNAPSHOT-jar-with-dependencies.jar /path-to-project/master.conf
```

سپس به گره‌های Worker بروید و دستور زیر را اجرا کنید.

```
java -jar worker-1.0-SNAPSHOT-jar-with-dependencies.jar /path-to-project/worker.conf
```

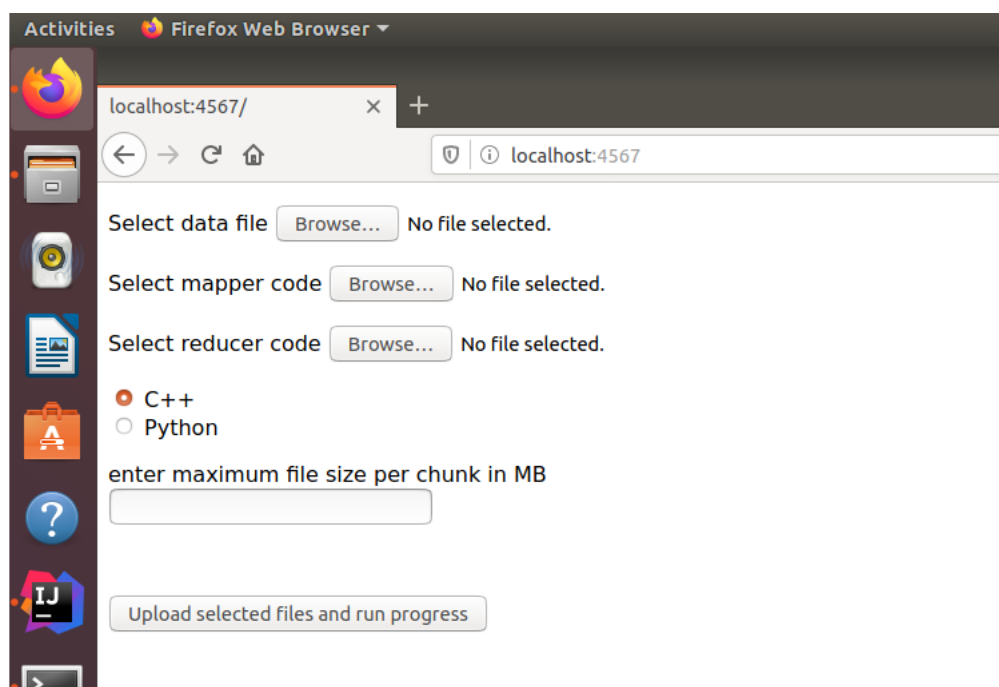
دقت کنید که فایل worker.conf برای هر worker باید جداگانه تنظیم شود.

حال به با مرورگر خود به لینک زیر بروید و از سامانه استفاده کنید.

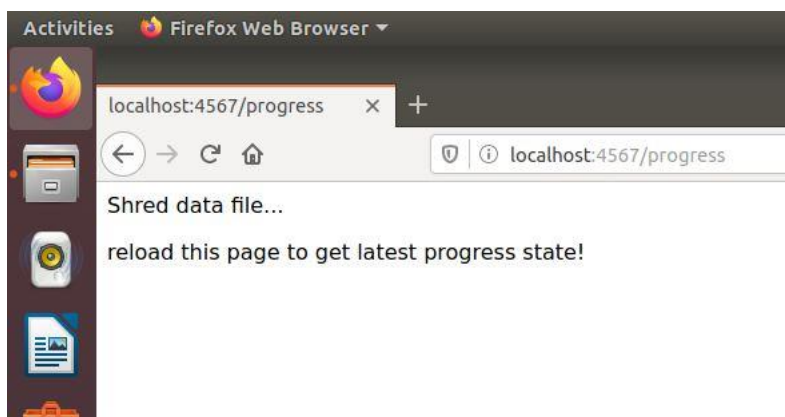
<http://master-ip-address:4567/>

پس از آپلود کردن فایل داده، Code های mapper و reducer، مشخص کردن زبان مورد استفاده و سبک هر فایل شکسته شده با کلیک بر روی کلید مربوطه پردازش بر روی بستر خوشه‌ای اجرا می‌شود که شما می‌توانید مراحل اجرای فرآیند را در مرورگر خود مشاهده کنید. (در ادامه عکس‌هایی از این روند آمده است)

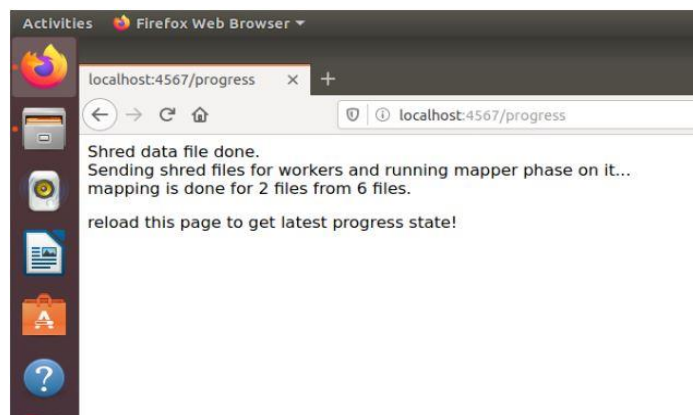
مراحل اجرای یک فرآیند به شرح تصویر:



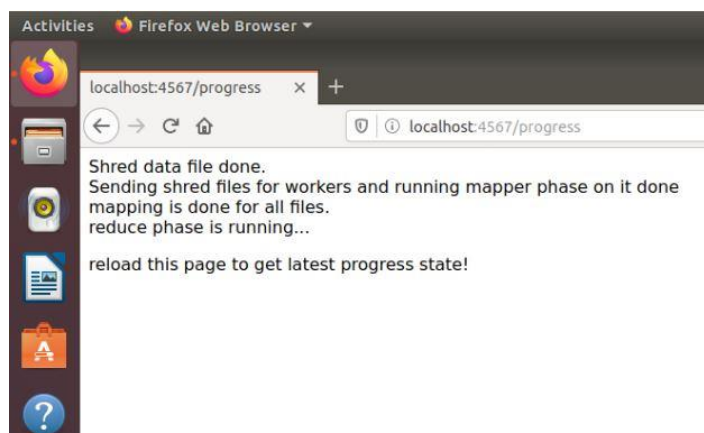
رسم توضیحی ۱ - مرحله‌ی انتخاب فایل داده و کد ها



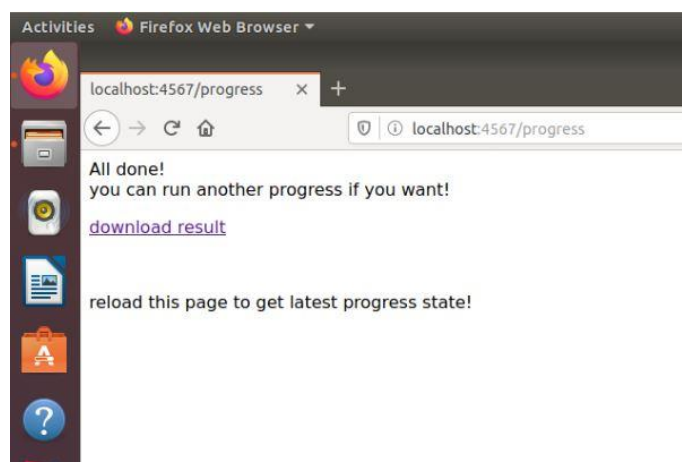
رسم توضیحی ۲ فاز تقسیم کردن فایل اصلی



رسم توضیحی ۳ فاز انجام پردازش در Worker ها



رسم توضیحی ۴ فاز اجرای کد جمع بندی در Master



مثال‌هایی از پردازش خوشه‌ای که روی این سامانه اجرا شده است.

چهار مثال شامل دو نمونه به زبان Python و دو نمونه به زبان Cpp در این سامانه اجرا شده‌است که ورودی، خروجی و کدهای هر دو مرحله پردازش را در زیر شاخه‌ی exampleCodes می‌توانید مشاهده کنید.

مثال اول: Word Count

این مثال که جز ساده ترین مثال‌های این پروژه است در دو مرحله تعداد کلمات تکرار شده در یک فایل داده را شمارش می‌کند.

مثال دوم: Character Count

در این مثال که بسیار به مثال قبلی شبیه است، در واقع در دو مرحله سامانه تعداد حروف ظاهر شده در یک فایل داده را شمرده و در فایل نتایج نهایی ذخیره می‌کند.

مثال سوم: شناسایی مزنونین

در این مثال فرض کنید که به ازای هر فرد داریم که در چه سالی به چه شهرهایی سفر کرده است و می‌خواهیم افرادی را شناسایی کنیم که رفتار عجیبی در یک سال داشته‌اند، این رفتار عجیب اینگونه تعریف می‌شود که یک فرد در یک سال به بیش از ده شهر سفر کرده باشد، در واقع در این مثال اسامی افرادی را پیدا می‌کنیم که در یک سال به بیش از ده شهر سفر داشته‌اند، در خروجی علاوه بر اسم شخص مزنون شهرهایی که این فرد به آن‌ها سفر کرده است هم می‌آید.

مثال چهارم: پیدا کردن بیشینه کاربر همزمان اپراتورهای تلفن همراه

در این مثال فرض کنید به ازای هر شماره موبایل تمامی بازه‌های زمانی‌ای که این شماره موبایل در یک روز بخصوص در حال مکالمه بوده‌است را در اختیار داریم، حال می‌خواهیم بیشینه تعداد کاربران در حال مکالمه در یک واحد زمانی به ازای سه اپراتور همراه اول، ایرانسل و رایتل را به دست آوریم که در واقع برابر پیک استفاده از آن اپراتور در طول یک روز می‌باشد.