

به نام خدا

مبانی رایانش ابری (نیمسال دوم تحصیلی ۹۹-۴۰۰)

پروژه پایانی

Single-Node Container Scheduling

آخرین تاریخ اپلود پاسخ در courses:

۱۴۰۰/۰۴/۲۸ (این زمان حقیقتاً قابل تمدید نیست)

مقدمه

در کلاس با مفهوم container و به تبع آن با Docker آشنا شدید. در پروژه پایانی درس، شما به شکل عملی با ابزار Docker آشنا می‌شوید. ویدئویی برای آشنایی با Docker توسط آقای فراهانی در سایت درس قرار داده شده است. شما همچنین می‌توانید تعداد زیادی راهنمای آنلاین حتی به زبان فارسی پیدا کنید (مثلاً در مکتب‌خونه).

زیرساخت:

مرحله اول: شما در این پروژه به Docker و دانستن نحوه ارسال دستورات به آن، چه از طریق CLI و چه از طریق روش‌های دیگر نیاز دارید (بسته به انتخاب شما). توصیه تیم درس این است که ابتدا یک ماشین مجازی (ubuntu server 20.04 (amd64 با حافظه اصلی و دیسک کافی بالا بیاورید و به آن تعداد ۴ هسته مجازی CPU یا همان vCPU اختصاص دهید (اگر لپ‌تاپ هر دو عضو گروه مشکل تعداد کم هسته CPU را دارید به sajavadi@aut.ac.ir ایمیل بزنید). شما Docker را می‌توانید به راحتی بر روی Ubuntu نصب کنید ولی محدود به Ubuntu نیستید و می‌توانید از هر سیستم عاملی دیگری یا سیستم عامل خود لپ‌تاپ استفاده کنید.

مرحله دوم: پس از نصب Docker سعی کنید با اجرای مثال‌هایی با نحوه کار کردن Docker و دیگر ابعاد آن آشنا شوید.

مسئله بالقوه: اگر بواسطه تحریم شدن آدرس‌های IP ایران توسط DockerHub به مشکل خوردید، سعی کنید از VPN استفاده کنید و اگر مشکل کماکان حل نشد، در سایت درس سوال را مطرح کنید.

مرحله سوم: حال نوبت به ساخت یک Docker image است که بایستی توسط شما ساخته شده (نوشتن Dockerfile و دیگر مراحل) و در درون این image بایستی که دستورهای زیر تعریف شوند پایه (base)

این image هر سیستم عاملی که شما می‌خواهید می‌تواند باشد و این دستورات را با زبان برنامه‌سازی دلخواه خود پیاده کنید. تنها محدودیت شما این است که پیاده‌سازی این دستورات بایستی به شکل single-threaded باشد). دقت کنید که الگوی مشابه تمامی این دستورات انجام یک عملیات مشخص بر روی محتوای یک فایل ورودی و نوشتن خروجی در یک فایل مشخص است.

نام دستور	ورودی	خروجی
min	یک فایل عددی با یک ستون	مینیمم اعداد موجود در فایل ورودی
max	یک فایل عددی همانند بالا	ماکزیمم اعداد موجود در فایل ورودی
average	یک فایل عددی همانند بالا	میانگین اعداد موجود در فایل ورودی
sort	یک فایل عددی همانند بالا	لیست مرتبط شده کاهشی از اعداد موجود در فایل ورودی
wordcount	یک فایل متنی	یک فایل با دو ستون که تعداد تکرار کلمات را نشان داده جایی که بر اساس تعداد تکرار از زیاد به کم مرتبط شده‌اند.

مرحله چهارم: بعد از ساختن Docker image مرحله قبلی، یک Container از این image بسازید و تک دستورات را تست کنید و از درستی عملکرد آنها اطمینان حاصل کنید.

مرحله پنجم: در مرحله پایانی زیرساخت شما بایستی که سه container از image مرحله سوم بسازید که همیشه در حال اجرا هستند و از آنها به عنوان pool of containers استفاده کنید. قاعدتا برنامه شما که در گام بعدی می‌نویسید بایستی بتواند کانتینرهای موجود در سیستم را اسکن کند و همیشه از سلامت این سه کانتینر اطمینان حاصل کند و شناسه کانتینرها (container ids) را در اختیار دیگر بخش‌های برنامه قرار دهد.

طرح اصلی پروژه

در این پروژه شما یک برنامه چند نخه را می‌نویسید که درخواست‌های کاربر را می‌گیرد و با توجه به ظرفیت خالی سیستم (تعدادی از آن سه کانتینر که کاری انجام نمی‌دهند)، دستور خواسته شده را در درون کانتینر بر روی فایل ورودی اجرا می‌کند و فایل خروجی را در اختیار کاربر قرار می‌دهد. برنامه شما بایستی حداقل از دو ریسمان تشکیل شده باشد: (۱) ریسمانی که درخواست‌های کاربر را دریافت و آنها را در انتهای یک صف قرار می‌دهد (دقت کنید که کاربر بایستی بتواند درخواست‌های خود را بدون انتظار برای درخواستی‌های قبلی به برنامه شما ارسال کند) و (۲) ریسمانی که یک درخواست را از ابتدا صف برداشته و با توجه به وضعیت سه کانتینر موجود در سیستم، انجام آن درخواست را شروع می‌کند. پیشنهاد ما این است که یک command line interface در اختیار کاربر قرار دهید ولی مطمئن باشید که این واسط بعد از دریافت یک درخواست کاربر و تخصیص یک شناسه یکتا به آن، بلافاصله برای دریافت درخواست بعدی اقدام می‌کند و بعد از آماده شدن خروجی، تنها به اطلاع کاربر می‌رساند که پاسخ درخواست با شناسه مشخص آماده شده و همه فایل‌های

خروجی در پوشه‌ای که توسط خود کاربر در متن درخواست مشخص شده است، قرار گرفته‌اند. پیش از توضیح شکل زیر به قالب درخواست کاربر توجه فرمایید:

Request format: {<operation name, input file path>, <operation name, input file path>, ..., <output directory>}

Example request: {<min, /tmp/grade.txt>, <max, /tmp/grade.txt>, </tmp/gradeStat>}

Example output (files within gradeStat):

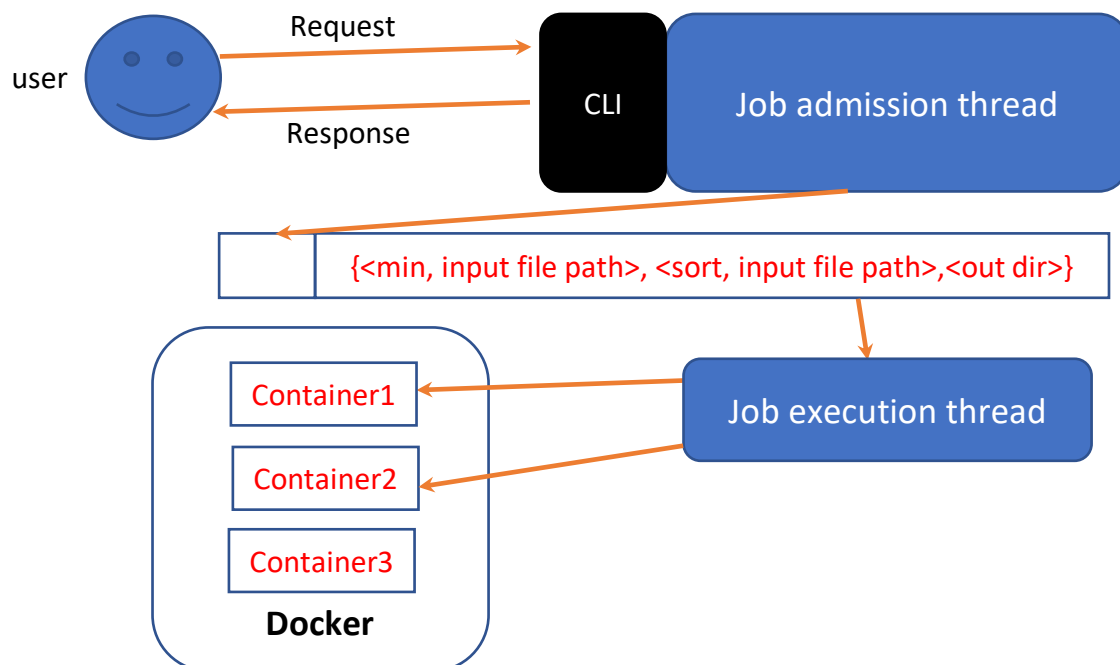
```
$ cd /tmp/gradeStat
```

```
$ ls
```

```
min.txt
```

```
max.txt
```

همانطوری که می‌بینید هر درخواست کاربر شامل یک یا چند دستور و فایل‌های ورودی متناظر آنها (تمام فایل‌ها در سیستم عامل میزبان هستند که در این پروژه می‌شود ماشین مجازی) و در نهایت پوشه‌ای در سیستم عامل میزبان که فایل‌های خروجی باید در آن قرار گیرند. شما بایستی که ورودی کاربر را در یک قالب یک فایل با همان الگوی بیان شده در مثال بالا، دریافت کنید. همچنین دقت کنید که اسم فایل‌های خروجی دستورات بیان بایستی برابر با نام دستور درخواستی باشد و یک دستور حداکثر یکبار در یک درخواست ظاهر می‌شود. شکل زیر یک نمای کلی از برنامه مد نظر پروژه را نشان می‌دهد.



همانطوری که در شکل بالا نمایش داده شده است، یک ریسمان از برنامه شما مسئول دریافت درخواست‌های کاربر و قرار دادن آن در انتهای صف محاسباتی است. هسته اصلی برنامه شما ریسمانی است که با خالی شدن هر کانتینر (در ابتدا هر سه کانتینر خالی هستند) بایستی که یک دستور از درخواست فعلی

(undergoing request) را پردازش کند یا اگر آخرین دستور درخواست فعلی در حال اجرا است، شروع به پردازش درخواست جدید موجود در ابتدای صف بکند. نکته اصلی این است که برنامه شما بایستی همیشه از وضعیت هر سه کانتینر مطلع باشد و در صورت موجود یک کانتینر خالی و درخواست‌های که هنوز پردازش نشده‌اند، یک دستور ارسالی را به ترتیب بیان شده بر روی کانتینر اجرا کند و خروجی را در پوشه درخواستی کاربر قرار دهد. بعد از تکمیل همه دستورات موجود در یک درخواست کاربر، کاربر بایستی از طریق CLI متوجه اتمام آن درخواست شود.

نمره اضافه

بخش امتیازی پروژه شما گرفتن فایل یک یا چند برنامه و اجرای آن برنامه(ها) بر روی کانتینر و در اختیار قرار دادن خروجی در قالب یک فایل است (نام فایل خروجی `program-name.out` بایستی باشد). این برنامه می‌تواند به زبان C++ باشد که پسوند `cpp` خواهد داشت و شما بایستی که ابتدائاً آن را در درون کانتینر کامپایل کنید و حتی المقدور سعی کنید کتابخانه‌های ناموجود را به درون کانتینر دانلود کنید. همچنین این برنامه می‌تواند به زبان پایتون باشد که همانند یک مورد قبلی بایستی سعی کنید کتابخانه‌های استفاده شده را به درون کانتینر دانلود کنید.

Request format: {<program name, input file path>, <program name, input file path>, ..., <output directory>}

دقت کنید که بعد از اجرای درخواستی از نوع بالا، بایستی که کانتینر را تمیز (clean) کنید تا یک برنامه اجرایی نتواند محیط یک کانتینر را برای برنامه‌های بعدی آلوده کند.

نحوه تحویل پروژه پایانی

۱. یکی از اعضای گروه، موارد زیر را در قالب یک فایل زیپ با نام «group_id_student_id1_student_id2_final_project» در صفحه درس اپلود کند. شماره گروه را از فایل اکسلی که برای تشکیل گروه‌ها استفاده شد، بازیابی کنید.

• گزارش که بایستی شامل پاسخ به بخش اول و گزارش انجام گام‌های مختلف بخش دوم و بخش امتیازی باشد. گزارش شما بایستی که از کیفیت خوب برخوردار بوده و از تکرار یا بی نظمی پرهیز کنید. *اولین بخش در گزارش جدولی است که تعیین می‌کند هر عضو گروه چه کارهایی را انجام داده است.* این تقسیم کار بایستی در زمان تحویل حضوری قابل راستی آزمایی باشد.

۲. دستیاران آموزشی علاوه بر بررسی گزارش‌ها و کدهای برنامه، از طریق اسکایپ، پروژه را به صورت اجرای زنده از شما تحویل خواهند گرفت. بنابراین بسیار مهم است که هر دو عضو گروه به پروژه تسلط داشته باشند. تسلط هر دو عضو گروه در ارائه نقش مهمی در نمره نهایی شما خواهد داشت. انتظار می‌رود عضوی

از یک گروه که تسلط بیشتری به این پروژه دارد، با همکاری موثر و کمک به عضو دیگر، نقش مهمی در آموزش جمعی حاصل از این کلاس، ایفا کند.

جریمه دیرکرد

با توجه به ددلاین نهائی کردن نمرات، هیچ پروژه‌ای بعد از ساعت ۲۳:۵۹ روز ۲۸ تیر ۱۴۰۰ پذیرفته نخواهد شد.

جریمه تقلب

۱. همه گروه‌ها بایستی که خود پروژه را انجام دهند و هرگونه تقلب یا ارسال کار دیگران یا کارهای موجود در وب که بخش برنامه‌نویسی پروژه را به شکل جزئی یا کلی انجام داده است، غیرقابل پذیرش و عواقب شدیدی خواهد داشت. دانشجویان بی شک می‌توانند از راهنمای موجود در وب یا کتابخانه‌های کمکی استفاده نکنند تا جایی که همه منابع و کتابخانه‌ها کمکی به صراحت ذکر شده باشد.
 ۲. بنده و گروه حل پروژه تمام تلاش خود را برای شناسایی تقلب‌های احتمالی خواهیم کرد تا در نهایت یک ارزیابی عادلانه از همه دانشجویان عزیز داشته باشیم. ما از MOSS برای شناسایی فایل‌های مشابه استفاده خواهیم کرد.
- در نهایت، هرگونه سوال در مورد پروژه و بخش‌های آنها را تنها و تنها از طریق سایت درس و ایجاد مباحثه با عناوین مرتبط مطرح بفرمایید. البته به بنده یادآوری کنید اگر سوال را ندیدم.

تندرست و موفق باشید

تیم درس مبانی رایانش ابری