



دانشکده‌ی مهندسی کامپیوتر

شبکه‌های کامپیوتری

مدرس: کامبیز میزانیان

تمرین عملی سری دوم

پس از انجام تمرین یک ویدیوی ۲ دقیقه‌ای ضبط کنید و در آن به صورت کلی روند حل سوال رو توضیح دهید. توجه کنید که نیاز به توضیح تمام جزئیات به صورت دقیق نیست و این ویدیو به منظور نشان دادن تسلط شما بر روی تمرین انجام شده است. شما در این تمرین مجاز به استفاده از هر زبان برنامه‌نویسی مانند جاوا، پایتون و ... هستید اما توصیه می‌کنیم از زبان پایتون استفاده کنید.

۱ Load Balancer

در این تکلیف، شما باید یک Load Balancer ساده را به کمک برنامه‌نویسی سوکت طراحی و پیاده‌سازی کنید که ترافیک شبکه ورودی را بین چندین سرور توزیع کند. یک Load Balancer نقش اساسی در بهبود عملکرد و قابلیت مقیاس‌پذیری برنامه‌های وب با توزیع بهینه درخواست‌های ورودی ایفا می‌کند. Load Balancer شما باید قادر باشد درخواست‌ها را بین چندین سرور پشتیبان توزیع کند.

۱.۱ الگوریتم توزیع بار

در این تمرین از الگوریتم Round Robin برای توزیع بار بین سرورهایی که پشت Load Balancer هستند استفاده کنید.

۲.۱ بررسی وضعیت سلامت سرورهای پشتیبان

با توجه به اینکه هر لحظه ممکن است هر کدام از سرورهای پشتیبان از دسترس خارج شوند می‌بایست هرچند دقیقه یک بار (انتخاب این بازه به عهده خودتان است) تمامی سرورهای پشتیبان با ارسال یک ریکوئست آزمایش شوند و در صورتی که بالا نبودند **موقتاً** از لیست سرورهای پشتیبان حذف شوند و تا دوره بعدی آزمایش سرورها، هیچ ریکوئستی به آن‌ها هدایت نشود.

۳.۱ توزیع بار

بعد از انجام مراحل قبلی نوبت به قسمت اصلی وظیفه Load Balancer می‌رسد، به ازای هر درخواست می‌بایست Load Balancer یک کانکشن با سرور پشتیبان برقرار کند و درخواستی که دریافت کرده است را به سرور بفرستد و پاسخ سرور را به کلاینت برگرداند.

۴.۱ کانفیگ Load Balancer

Load Balancer شما می‌بایست از فایل ای‌کونفیگ که کنارش قرار می‌گیرد لیست سرورهای پشتیبان را استخراج کند و از آن‌ها استفاده کند. فرمت فایل config به شکل زیر است.

```
[ ip0 ] [ port0 ]
[ ip1 ] [ port1 ]
[ ip2 ] [ port2 ]
```

آزمایش Load Balancer

جهت سادگی آزمایش یک سرور ساده را نیز پیاده‌سازی کنید که با دریافت یک عدد اجرا می‌شود و به درخواست‌های ورودی پاسخ زیر را می‌دهد:

It's instance number <i>

توجه کنید که *i* همان عددی است که ابتدا در ورودی به برنامه داده‌ایم، همچنین هر instance از سرویس می‌بایست پورت تصادفی‌ای که روی آن بالا آمده است را به همراه آیدی در ابتدای اجرا شدن نمایش دهد.

نکته مهم: توجه کنید که نیاز است در برنامه‌ی Load Balancer به ازای هر ریکوئستی که به این سرویس ارسال می‌شود باید آدرس مبدا و سروری که آن ریکوئست به آن ارسال می‌شود به عنوان لاگ نمایش داده شود. همچنین در برنامه‌ی سرور نیز باید لاگی که نمایش دهنده‌ی پاسخ به یک ریکوئست است نمایش داده شود.

ارزیابی

جهت تست Load Balancer شما، دستیاران آموزشی چند instance از سرویسی که ارائه کرده‌اید بالا می‌آورند و همچنین Load Balancer شما را هم اجرا می‌کنند، سپس به مقدار خیلی زیاد درخواست به سمت Load Balancer شما ارسال می‌شود و نتایج درخواست‌ها بررسی می‌شود، در نهایت می‌بایست میان نتایج درخواست‌ها به تعداد **تقریباً** مساوی از هر instance وجود داشته باشد.

تحويل تمرین

فایل ارسالی شما می‌بایست به فرمت زیر باشد:

```
[ student_number ]_PRACTICAL_2.zip
hash.txt // sha256 of your video file
server.py // or any other format
serveice.py // or any other format
README.md // anythong about running your code?
url.txt // url to your video file
```

نکته: همچنین می‌توانید ویدیو رو به صورت مستقیم در کنار کدها آپلود کنید.

موفق باشید