

به نام خدا

مبانی رایانش ابری (نیمسال اول تحصیلی ۹۹-۴۰۰)

پروژه پایانی

Elastic Load Balancing via Haproxy

آخرین تاریخ آپلود پاسخ در courses:

۱۳۹۹/۱۱/۱۷

بخش ۱

برای این بخش، ابتدا مسئله توزیع بار را تعریف کنید. یکی از توزیع کننده‌های متن باز پر استفاده HaProxy^۱ است. کوتاه در مورد HaProxy توضیحاتی ارائه کنید و سپس الگوریتم‌های توزیع بار پشتیبانی شده در HaProxy را به شکل خلاصه توضیح دهید. دقت کنید که پروژه بر روی HaProxy 2.0.13 خواهد بود.

```
ahmad@vm1:~$  
ahmad@vm1:~$ apt show haproxy  
Package: haproxy  
Version: 2.0.13-2ubuntu0.1
```

بخش ۲

این بخش به زیر بخش‌های زیر تقسیم می‌شود (کماکان از VirtualBox استفاده می‌کنیم).

۱-۲: ابتدا یک ماشین مجازی Ubuntu server 20.04 با مشخصات (1 CPU core, 512GB Ram, 16GB disk space) ایجاد کنید (VM1) و یک web server بر روی آن راه اندازی کنید. انتخاب خدمتگزار وب به عهده شما است و هر خدمتگزار که از http پشتیبانی می‌کند قابل قبول است. توصیه ما apache یا nginx است. نشان دهید که صفحه پیشفرض وب سرور از مرورگر سیستم میزبان قابل دسترسی است. همچنین با ابزار collectd آشنا شوید. آن را بر روی VM1 نصب کنید و با پیکربندی مناسب آن، کاری کنید که میزان منابع استفاده شده در VM1 (برای CPU و Ram) در واسط تحت وب collectd نمایش داده شود. راهنمای مختلف برای اینکار وجود دارد. به عنوان مثال:

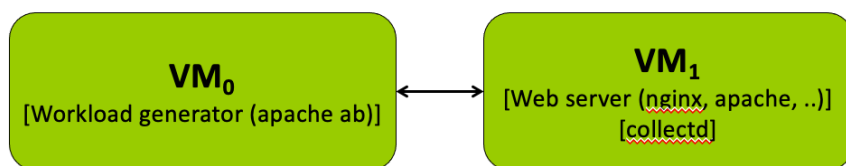
<https://www.linuxsysadmins.com/install-collectd-monitoring-on-linux/>

شما همچنین می‌توانید collectd را به گونه‌ای پیکربندی کنید که خروجی خود را در قالب فایل CSV بنویسد. به این ویژگی در ادامه کار نیاز خواهید داشت. به CSV plugin برای collectd رجوع کنید.

۲-۲: سه فایل در ماشین مجازی ایجاد کنید: (۱) یک فایل که دسترسی به آن CPU-intensive است (مثلا cpu-intensive.php)، (۲) یک فایل که دسترسی به آن memory-intensive است (مثلا memory-intensive.php) و (۳) فایل دیگری که دسترسی به آن هر دو ویژگی CPU-intensive و memory-intensive بودن را دارد (مثلا cpu-mem-intensive.php). به این فایل‌های از طریق مرورگر دسترسی پیدا کنید و screenshot را در گزارش خود بگنجانید.

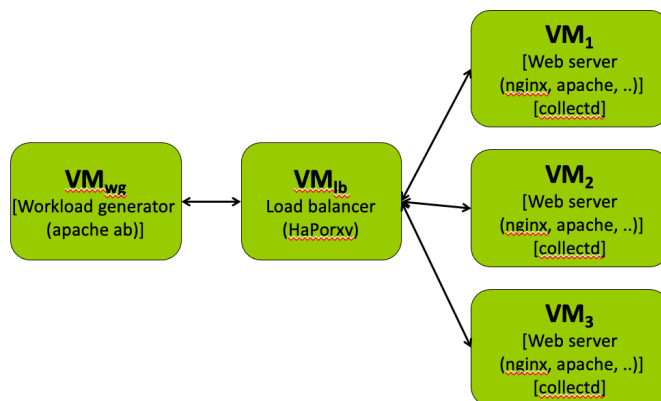
¹ <http://www.haproxy.org/>

۲-۳: برای تست کردن نتیجه مرحله ۲-۲، ماشین مجازی دیگری ایجاد کنید (clone) و با اجرای یک workload generator (توصیه ما apache ab است) و ارسال درخواست به فایل‌های ایجاد شده در مرحله ۲-۲، نشان دهید که منابع استفاده شده در VM₁ با فایل‌هایی که به آنها درخواست فرستاده شده است، تناسب دارد. اطلاعات نمایش داده شده توسط collectd را در گزارش خود بگنجانید. بنابراین تا این مرحله بایستی که معماری زیرین را داشته باشید.



۲-۴: در این مرحله بایستی نمودار throughput (تعداد درخواست‌های تکمیل شده در برابر تعداد درخواست‌های ارسال شده) ماشین مجازی VM₁ را برای سه فایل مذکور بدست بیاورید. بدین منظور workload generator را با نرخ‌های افزایشی اجرا کنید (دقت کنید که هر آزمایش حداقل چند دقیقه بایستی باشد). حالا نمودار نرخ واقعی (actual rate in y axis) را بر اساس نرخ اسمی (rate that you set in apache ab or another workload generator) رسم کنید. (نرخ اسمی در محور x). این نمودار و تحلیلی از آن را در گزارش خود بگنجانید.

۲-۵: حال حداقل ۲ ماشین مجازی دیگر همانند VM₁ ایجاد کنید (clone). همچنین یک ماشین مجازی دیگر همانند VM₀ ایجاد کنید و HaProxy را بر روی آن نصب کنید. HaProxy را پیکربندی کنید تا از خط‌مشی round-robin استفاده کند. در گزارش خود نشان دهید که این معماری کار می‌کند. بدین مذکور به فایل‌های ایجاد شده از طریق مرورگر دسترسی پیدا کنید اما این بار از ادرس IP توزیع کننده بار (load balancer) استفاده کنید. ۳ درخواست برای هر کدام از فایل‌ها مرحله ۲-۲ بفرستید و با مشاهده apache log (web server log) در هر کدام از VM₁₋₃ ببینید کدام VM برای پردازش درخواست ارسال شده به توزیع کنند بار انتخاب شده است.



۲-۶: مرحله ۲-۴ مجدداً تکرار کنید و نمودار throughput را اینبار برای load balancer رسم کنید. چه تفاوتی بین این نمودار و نمودار مرحله ۲-۴ وجود دارد؟

۲-۷: با توجه به تعداد درخواست‌های ارسال شده به load balancer، ممکن است نیازی به روشن بودن ۳ ماشین مجازی نباشد. در این بخش بستر نرم افزاری را ایجاد می‌کنید که VM₂ و VM₃ را بر اساس میزان درخواست‌های ورودی و میزان استفاده از منابع در backend VMs روشن یا خاموش می‌کند (به شکل پویا در طول زمان). دقت کنید که VM₁ بایستی همیشه روشن باشد. بستر نرم افزاری (software demean) ایجاد کنید (بخشی از این بستر در backend VM بخشی در load balancer و بخشی از آن در host اجرا می‌شود). این بستر نرم افزاری بایستی که قواعد زیر را پیاده کند (سیستم شما با VM₁ به صورت روشن و VM₂ و VM₃ به صورت خاموش، کار خود را آغاز می‌کند). دقت کنید تنها خاموش کردن ماشین‌های مجازی کافی نیست و پیکربندی HaProxy نیز بایستی هر بار تغییر کنید و دستور reset اجرا شود.

- اگر میانگین CPU usage برای VM₁ در ۲ دقیقه گذشته از 80% بیشتر است، VM₂ روشن شود.

- اگر میانگین CPU usage برای VM₁ و VM₂ در ۲ دقیقه گذشته از ۸۰٪ درصد بیشتر است، VM₃ روشن شود. فرض می‌کنیم بیشتر از ۳ ماشین مجازی برای backend نمی‌توان داشت.
 - اگر میانگین CPU usage برای VM₁، VM₂ و VM₃ در ۲ دقیقه گذشته از 50٪ کمتر است، VM₃ خاموش شود.
 - اگر اگر میانگین CPU usage برای VM₁ و VM₂ در ۲ دقیقه گذشته از ۴۰٪ کمتر است، VM₂ خاموش شود.
- در گزارش خود عملکرد موفقیت‌آمیز این بستر نرم‌افزاری را با استفاده از آزمایش‌های مختلف نشان دهید و برای ارائه پروژه خود آماده باشید. دقت کنید که می‌توانید بر روی VM_{wg} چند نسخه از workload generator را اجرا کنید (e.g., using screen) و با کم و اضافه کردن این نسخه‌ها، بار دلخواه خود را برای load balancer ایجاد کنید.

نمره اضافه

- در بخش دوم خط‌مشی خاموش و روشن کردن ماشین‌های مجازی تنها بر اساس CPU usage بود. اگر یک خط‌مشی ترکیبی پیاده کنید که memory usage را هم دخیل کند که بتوان وزن‌هایی را برای CPU usage و memroy usage پیکربندی کرد و همچنین بازه زمانی ۲ دقیقه قابل پیکربندی باشد، نمره اضافی را خواهید گرفت.

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 اکسل 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 بی‌نظمی 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 راستی‌آزمایی 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

هر روز تاخیر در ارسال پروژه ۱۰٪ نمره منفی خواهد داشت. امکان اپلود پروژه تنها تا ۳ روز از تاریخ تعیین شده ممکن خواهد بود.

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 بی‌شک 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.
۳. در صورت شناسایی تقلبی که ۵۰٪ یا پایین‌تر از کار را شامل می‌شود، اعضای گروه یک کارت زرد دریافت کرده و نمره «۰.۵-» * بارم پروژه» به اعضای گروه تعلق می‌گیرد و در صورت شناسایی تقلبی که بیشتر از ۵۰٪ کار را پوشش می‌دهد به اعضای گروه کارت قرمز تعلق گرفته و نمره «۱-» * بارم پروژه» به اعضای گروه تعلق می‌گیرد. علاوه بر این نمره منفی، گرفتن چند کارت زرد یا قرمز، عواقب شدیدتری خواهد داشت. 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.

تندرست و موفق باشید

تیم درس مبانی رایانش ابری