

به نام خدا

Resource Provisioning in Software Defined Networking Using Network Function Virtualization

واژه‌های کلیدی:

*Software Defined Networking, Resource Provisioning, Distributed Computing,
Network Function Virtualization*

پرهام الوانی (۹۲۳۱۰۵۸)

چکیده

در این مقاله قصد داریم برای بهبود تخصیص منابع در شبکه‌های نرم‌افزار بنیان از مجازی سازی توابع شبکه استفاده کنیم، در این روش تمامی قسمت‌های شبکه به صورت نرم‌افزاری پیاده‌سازی میشوند و برای بهبود زمان لازم برای بررسی بسته‌ها از کتابخانه‌هایی مانند DPDK استفاده میکنیم.

۱ مقدمه

ایده شبکه‌های قابل برنامه ریزی به تازگی با توجه به ظهور SDN شتاب قابل توجهی گرفته است. SDN وعده داده است که به صورت چشمگیری مدیریت شبکه را آسان کند و همچنین با استفاده از قابلیت برنامه نویسی شبکه، امکان پیاده سازی ایده‌های جدید در شبکه به سادگی را بوجود آورد. شبکه‌های نرم افزار بنیان با وجود اینکه مدت زمان زیادی نیست که مطرح شده‌اند ولی در صنعت نیز جای خود را پیدا کرده‌اند و روز به روز گسترده‌تر می‌شوند. چالش‌های بسیاری برای شبکه‌های نرم افزار بنیان وجود دارد که یکی از آنها نحوه تخصیص و آماده سازی منابع در این شبکه‌هاست. مساله تخصیص منابع در شبکه‌های نرم افزار بنیان یکی از مهمترین چالش‌های این زمینه تحقیقاتی می‌باشد. پلتفرم‌ها و بسترهای نرم افزاری مختلفی در این زمینه معرفی شده‌اند که در این تحقیق قصد داریم این بسترها و پلتفرم‌ها را بررسی و مقایسه نماییم و نشان دهیم که ایده‌ی مجازی‌سازی توابع شبکه در این امر چه تاثیری دارد. پیش از بررسی راه‌کار مورد نظرمون یعنی مجازی‌سازی توابع شبکه، به بررسی سایر راه‌کارهای موجود می‌پردازیم.

۲ کارهای مرتبط

۱.۲ Devoflow

Devoflow یک افزونه‌ای است که به OpenFlow اضافه شده و در شبکه‌های بزرگ با حجم ترافیک بالا استفاده می‌شود.

در این روش برای کاهش سربار flow-base switching از دو مکانیزم به نام‌های Rule cloning و Local actions استفاده می‌شود. در مدل استاندارد پروتکل OpenFlow اگر از قوانینی تحت عنوان wildcard استفاده کنیم تا تقاضاها به کنترلر را کاهش دهیم آنگاه تمام بسته‌هایی که با این قوانین هماهنگ هستند تحت عنوان یک Flow طبقه بندی می‌شوند. در مکانیزم اول devoflow بیان می‌شود که می‌توان در صورت نیاز قوانین wildcard را با پرچم CLONE مشخص کرده و به این ترتیب آنها را وادار کرد بسته‌های هماهنگ با این قوانین را به Flow های مختلف تقسیم کنند. برای این امر سوئیچ یک قانون جدید با مشخصات بسته ورودی هماهنگ با قانون wildcard ایجاد میکند و به این ترتیب سایر بسته‌های این Flow با قانون جدید مطابقت پیدا میکنند. مکانیزم دوم به زبان ساده بیان میکند که برای کاهش تقاضاها به کنترلر می‌توان تعدادی از تصمیم‌ها را در سمت سوئیچ گرفته و سربار کنترلر را کاهش داد. این روش به اینصورت عمل می‌کند که اگر Flow ورودی کوچک باشد از Devoflow استفاده می‌شود و اگر Flow بزرگ باشد از کنترلر استفاده می‌کند که این امر باعث میشود جریان‌های کوچک حافظه‌ای را اشغال نکنند و فضا را در اختیار جریان‌های بزرگتر قرار دهد.

مراجع

- [١] Patricia A. Morreale, *Software Defined Networking: Design and Deployment*, 6000 Broken Sound Parkway NW, CRC Press, 2015.
- [٢] Timothy Wood, *NetVM: High Performance and Flexible Networking Using Virtualization on Commodity Platforms*, 11th USENIX Symposium on Networked Systems Design and Implementation 2014.
- [٣] Eric Keller, *Stateless Network Functions*, HotMiddlebox'15, August 17-21, 2015, London, United Kingdom.
- [٤] H. Fei, H. Qi and B. Ke, *A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation*, IEEE Communication Surveys & Tutorials, vol. 16, no. 4, 2014.
- [٥] Thomas D. Nadeau, Ken Gray, *SDN: Software Defined Networks*, O'Reilly Media, 2013.