گزارش پروژه نهایی

شبکه‌های کامپیوتری پیشرفته

سپند سراج

سید سعید صفایی

مقدمه:

زنجیره سرویس‌ها در شبکه‌های کامپیوتری به مجموعه‌ای از سرویس‌ها گفته می‌شود که یک میزبان در زمان دریافت یک پکت از شبکه روی آن انجام می‌دهد تا بتواند آن را به درستی پردازش و سرویس‌دهی کند. از جمله سرویس‌های از این قبیل میتوان به Intrusion Detection System(IDS)، Firewall(FW) و Load Balancer(LB) اشاره کرد. این سرویس‌ها باید به صورت ترتیبی روی یک پکت اعمال شوند و به همین علت از نام زنجیره سرویس Service Function Chain(SFC) استفاده می‌کنیم.

به صورت تاریخی همیشه ارائه دهندگان سرویس‌های ابری که ارائه دهنده اصلی این سرویس‌ها بودند برای هر کدام از این سرویس‌ها از سخت‌افزار‌های خاص منظوره و بسیار هزینه‌بر استفاده می‌کردند که قابلیت مقیاس پذیری بسیار کمی داشتند و نیازمند سرمایه‌گذاری اولیه بسیار زیاد بودند.

با پیشرفت تکنولوژی و افزایش چشم‌گیر کارایی پردازنده‌های عام‌منظوره مفهوم مجازی سازی Virtualization توجه سازندگان این سیستم‌ها را جلب کرد. از بزرگترین مزیت‌های مجازی‌سازی سیستم‌ها می‌توان به افزایش مقیاس‌پذیری و کاهش هزینه‌ها اشاره کرد.

طراحی‌کنندگان این سیستم‌های مجازی برای تست و بررسی میزان کارایی نیازمند سیستمی جهت شبیه‌سازی طرح‌های خود بودند تا بتوانند قبل از پیاده‌سازی چنین سیستم‌های پیچیده و هزینه بری از کارایی سیستم خود اطمینان حاصل کنند. شبیه‌ساز CloudSim یکی از این شبیه‌سازها می‌باشد که تمامی ابزار و پارامتر‌های مورد نیاز جهت مطالعه و بررسی کارایی سیستم‌های زنجیره سرویس شبکه‌ای را برای طراحی کنندگان فراهم می‌کند.

مطالعه و بررسی مقاله

طوسی و همکاران [1] در سال ۲۰۱۹ مطالعه‌ای پیرامون تکنیک‌های مقیاس پذیری منابع برای زنجیره سرویس‌های شبکه‌ای ارائه انجام دادند. از دستاورد‌های این مطالعه ارائه روشی نوین جهت مانیتورینگ و مقیاس‌پذیری لحظه‌ای منابع سیستم برای ارائه زنجیره سرویس‌های شبکه‌ای می‌باشد. از نکات حائز اهمیت این مطالعه توجه به پارامتر‌های کارایی سیستم مانند تاخیر در پاسخگویی Lantency می‌باشد.

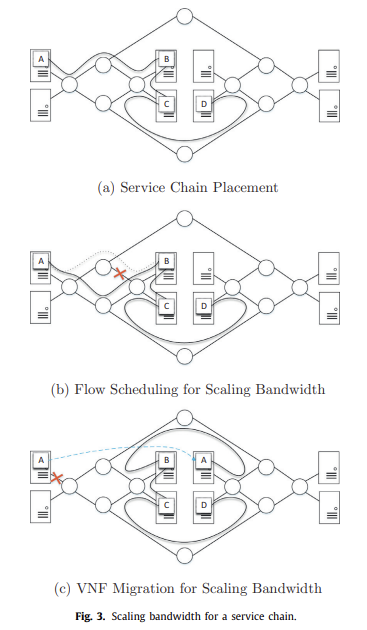
در طی این مطالعه فرض شده است که منابع اصلی که همواره مورد نیاز زنجیره سرویس‌دهی می‌باشند یکی پهنای باند Bandwidth و قدرت پردازش CPU می‌باشند. هرچند که الگوریتم‌های ارائه شده در این مطالعه قابل بسط دادن به دیگر منابع سیستمی نیز می‌باشند.

منابع فوق الذکر به دو طریق قابل مقیاس‌پذیری می‌باشند. اصطلاحا در مقیاس پذیری عمودی تعداد منابع ثابت می‌باشد و کارایی آن‌ها افزایش می باشد اما در مقیاس‌پذیری افقی تعداد منابع افزایش می‌یابند. یکی دیگر از نکات حائز اهمیت در این مطالعه ارائه راهکار‌های نوینی جهت پشتیبانی از هر دو نوع مقیاس‌پذیری به صورت لحظه‌ای در سیستم برای هر دو نوع منبع پهنای باند و قدرت پردازشی می‌باشد.

Diagram

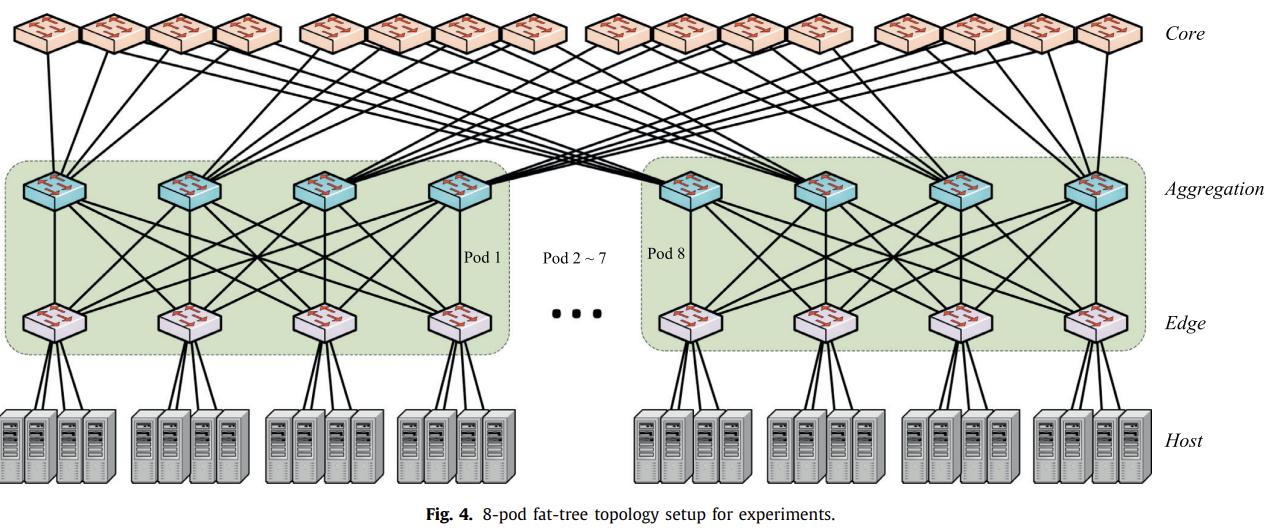
Description automatically generated

از دیگر تکنیک‌های ارائه شده در این مقاله جابجایی مکان سرویس‌دهی‌ها داخل شبکه می‌باشد که بعضا در راستای رفع محدودیت‌های پهنای باند با استفاده از منابع فعلی کمک می‌کند. نحوه عملکرد این تکنیک را می‌توانید در شکل زیر مشاهده کنید:

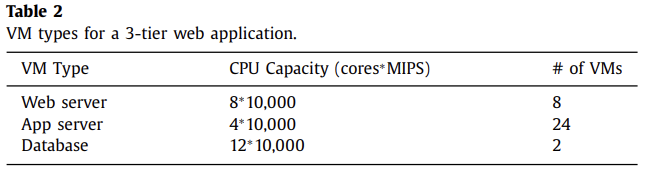


راه‌اندازی آزمایش

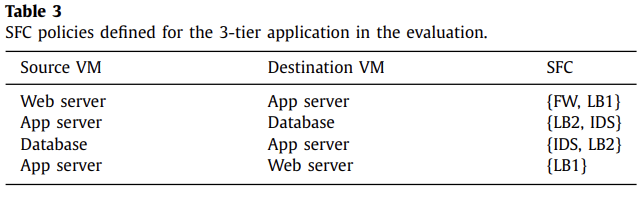
در راستای صحت سنجی یافته‌های این مطالعه و راه‌حل پیشنهادی آزمایشی طراحی شده است که ما در این پروژه سعی در پیاده‌سازی آزمایشی با پارامتر‌های مشابه داریم.  
ساختار فیزیکال این شبکه شبیه‌سازی شده از مدل fat tree 8 pod پیروی می‌کند که دارای ۳۲ سوییچ مرزی، ۳۲ سوییچ تجمیع کننده و ۱۶ سوییچ هسته‌ای می‌باشد که مجموعا ۱۲۸ واحد پردازش کننده‌ را به همدیگر متصل می‌کنند. ساختار پیشنهادی را می‌توانید در تصویر مشاهده کنید:



در ادامه ویژگی‌های واحد‌های پردازشی و انواع آنها تشریح شده‌اند که ما نیز محیط شبیه‌سازی خود را طبق همین جداول تنظیم کردیم.



در ادامه نیز سیاست‌های سرور‌های موجود در شبکه در برابر پکت‌های دریافتی از جهت زنجیره سرویس‌های ارائه شده توصیف شده‌اند.

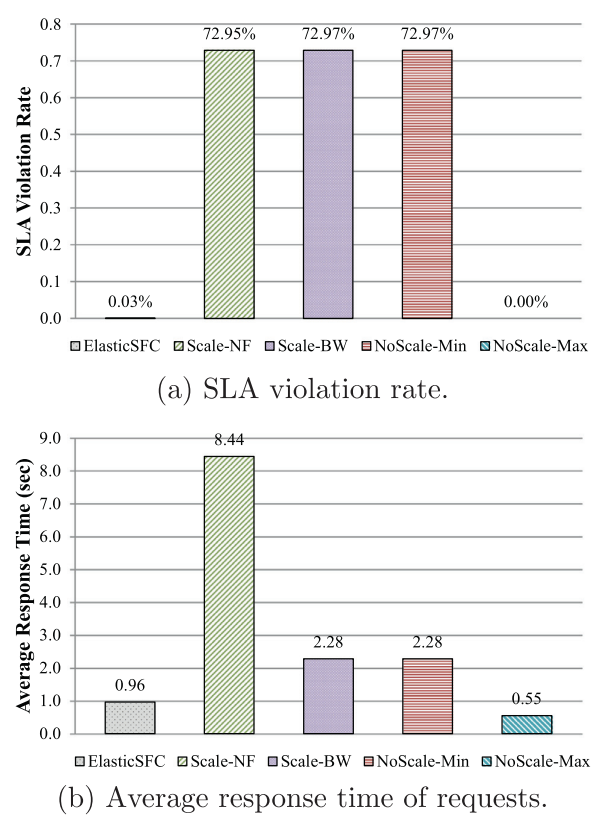


پس از طراحی‌ سیستم مورد نظر در فضای شبیه‌سازی نوبت به طراحی یک سناریو نزدیک به واقعیت در راستای تست میزان کارایی سیستم می‌رسد. برای این بخش نویسندگان مقاله از الگوی ترافیک وبسایت ویکیپدیا به زبان آلمانی طی یک بازه ۲۴ ساعته استفاده کرده‌اند که در مجموع در این بازه ۲۹ میلیون درخواست به سرور‌ها زده می‌شود. الگوی این ترافیک در شکل زیر مشاهده می‌شود.

Chart, line chart

Description automatically generated

در انتها نویسندگان الگوریتم پیشنهادی خود را با ۴ الگوریتم پایه دیگر (حداقل منابع، حداکثر منابع، مقیاس پذیری پهنای‌باند و مقیاس پذیری قدرت پردازش) مقایسه کردند و به خروجی‌های زیر رسیدند. در فصل ۲ بیشتر راجع به آنها توضیح می‌دهیم و سعی می‌کنیم این خروجی‌ها بازسازی ‌کنیم.



فصل ۲ گزارش کار پروژه

نصب Cloudsim-sdn

طبق توصیه تعریف پروژه نسخه cloudsimsdn که از شبیه‌ساز Cloudsim را انتخاب کردیم.

برای نصب این نسخه از شبیه ساز ابتدا سورس کد پروژه Cloudsim نسخه ۴ را از لینک ]۲[ دانلود می‌کنیم. همچنین Cloudsim-sdn را از لینک ]۳[ دانلود می‌کنیم. سپس فولدر

cloudsim-cloudsim-4.0\cloudsim-cloudsim-4.0\modules\cloudsim\src\main

از پروژه Cloudsim را داخل آدرس زیر از پروژه Cloudsim-sdn کپی می‌کنیم.

cloudsimsdn-master\cloudsimsdn-master\src\main

در این مرحله پروژه cloudsim-sdn کامل می‌باشد. و دیگر کاری با cloudsim نداریم.

در ادامه باید فایل pom.xml را بروزرسانی کنیم و خط زیر را در پایان وابستگی‌ها اضافه می‌کنیم.

<dependency>

<groupId>com.opencsv</groupId>

<artifactId>opencsv</artifactId>

<version>3.7</version>

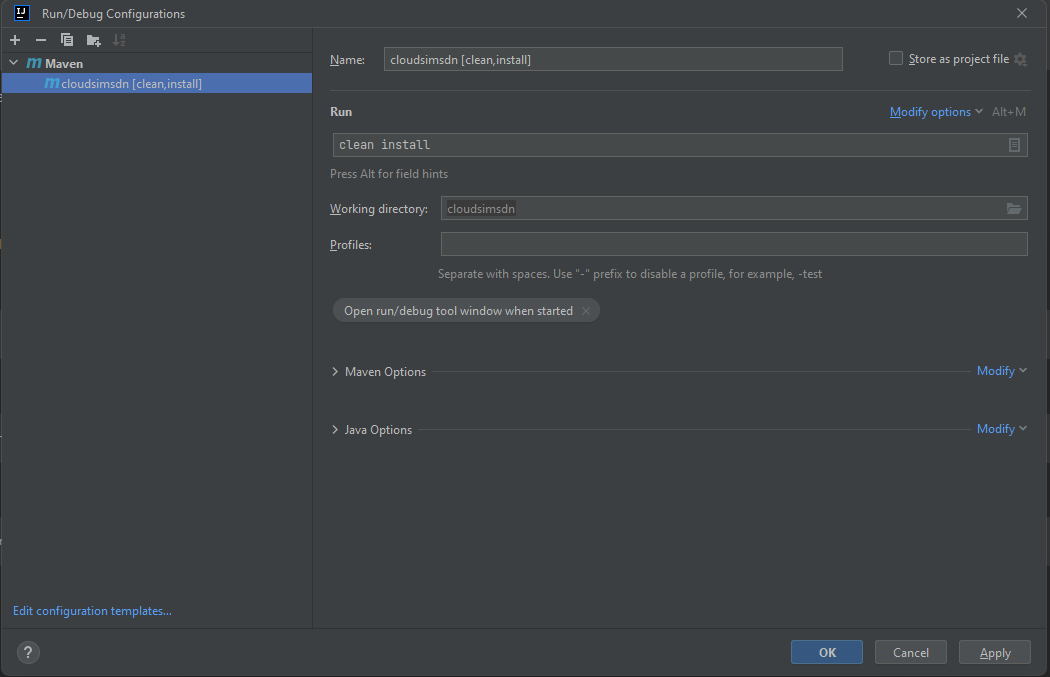
</dependency>

در این مرحله می‌توانیم به نصب و راه‌اندازی شبیه ساز بپردازیم.

راه اندازی محیط توسعه

در این پروژه برای توسعه پروژه از نرم‌افزار IntelliJ IDEA Community edition استفاده می‌کنیم.

پس از افزودن پوشه پروژه به برنامه باید از طریق ابزار maven پروژه را نصب کنیم. برای انجام این کار ابتدا یک پروفایل اجرا با مشخصات زیر می‌سازیم.



پس از افزودن با اجرای این دستور با موفقیت پروژه روی سیستم شما نصب می‌شود. 