مروری بر روش‌های مجازی‌سازی شبکه مرکز داده

ساناز کیانی‌فرد[1,*]، ادریس خضری[2]

[1,*]دانشجو، کارشناسی ارشد مهندسی فناوری اطلاعات گرایش شبکه‌های کامپیوتری، دانشگاه علوم و تحقیقات قزوین

[2] دانشجو، دکتری مهندسی نرم‌افزار، دانشکده برق، رایانه و فناوری اطلاعات، دانشگاه آزاد اسلامی قزوین

چکیده

با رشد حجم داده‌ها و تنوع برنامه‌های کاربردی اینترنتی، مراکز داده، زیرساخت کارا و امیدوار کننده‌ایی برای حمایت از ذخیره‌سازی داده و پلت فرمی برای استقرار سرویس‌ها و برنامه‌های کاربردی متنوع شبکه ارائه داده‌اند. این برنامه‌های کاربردی و سرویس‌ها غالباً تقاضاهای منابع متنوعی (فضای ذخیره‌سازی، قدرت محاسباتی، پهنای باند، تاخیر رفت و برگشت) را به زیرساخت تحمیل می‌کنند. معماری مراکز داده موجود از کمبود انعطاف‌پذیری برای پشتیبانی کارا از این برنامه‌های کاربردی رنج می‌برد در نتیجه پشتیبانی ضعیفی از کیفیت سرویس، قابلیت استقرار، قابلیت مدیریت و دفاع دربرابر حملات امنیتی ارائه می‌دهد. مجازی‌سازی شبکه مرکز داده رویکرد امیدبخشی برای رسیدگی به این مسائل است. مراکز داده مجازی‌شده برای ارائه بهترین انعطاف در مدیریت، هزینه کمتر، قابلیت توسعه، استفاده بهتر از منابع و کارایی انرژی پیش‌بینی شده‌اند. در این مقاله، ما روش‌های نوین فعلی در شبکه‌های مرکز داده را بررسی نموده و مقایسه‌ای دقیق از روش‌های بررسی شده ارائه داده‌ایم.

واژگان کلیدی: معماری مرکز داده، مجازی‌سازی، مرکز داده مجازی

1- مقدمه

مراکز داده اخیراً توجه زیادی را به خاطر زیرساخت کارا و کم هزینه برای ذخیره حجم وسیعی از داده‌ها و میزبانی از برنامه‌های کاربردی سرویس‌هایی با مقیاس بزرگ به خود جلب کرده‌اند. امروزه شرکت‌های بزرگ مثل آمازون، گوگل، فیس بوک و یاهو از مراکز داده برای ذخیره‌سازی، جستجوی وب و محاسبات در مقیاس بزرگ استفاده می‌کنند[6,13,27]. اما علی‌رغم اهمیت آنها، معماری مراکز داده امروزی هنوز از ایده آل خود خیلی دور هستند. به صورت سنتی، مراکز داده از سرورهای اختصاصی برای اجرای برنامه‌های کاربردی استفاده می‌کنند در نتیجه در بهره‌برداری از سرور ضعیف هستند و هزینه عملیاتی زیادی دارند. شرایط با ظهور فناوری‌های مجازی‌سازی سرور بهبود یافته است که اجازه می‌دهند چندین ماشین مجازی بر روی یک ماشین فیزیکی قرار بگیرند. این فناوری‌ها می‌توانند عملکرد ایزوله‌سازی را بین ماشین‌های مجازی یک سرور فیزیکی ارائه دهند که عملکرد برنامه‌های کاربردی را بهبود بخشیده و از حمله‌های اختلالی جلوگیری می‌کند. گرچه مجازی‌سازی سرور به تنهایی برای مقابله با تمام محدودیت‌های معماری مراکز داده امروزی کافی نیست.

اخیراً گرایشی در مجازی‌سازی شبکه‌های مراکز داده علاوه بر مجازی‌سازی سرور ایجاد شده است. هدف مجازی‌سازی شبکه‌ایجاد شبکه‌های مجازی چندگانه بر روی یک شبکه فیزیکی اشتراکی است [18] که به هر شبکه مجازی اجازه اجرا و مدیریت مستقل را می‌دهد. با جدا کردن شبکه‌های منطقی از زیرساخت شبکه فیزیکی، امکان معرفی پروتکل‌های شبکه شخصی‌سازی شده و سیاست‌های مدیریت، اجرای عملکرد ایزوله‌سازی و کاربردهای تامین کیفیت سرویس فراهم شده است.

همچنین، پیشنهاد ایزوله‌سازی در محیط‌های شبکه‌های مجازی می‌تواند تأثیر تهدیدهای امنیتی را نیز کمینه کند. مجازی‌سازی شبکه‌های مرکز داده جهت تحقیقاتی نسبتاً جدیدی است و یک گام کلیدی به سمت معماری‌های کاملاً مجازی‌شده مرکز داده است.

در حالیکه مجازی‌سازی شبکه‌های مرکز داده به تمام موضوعات بالا رسیدگی می‌کند، چالش‌های تحقیقاتی جدیدی را نیز شامل فناوری‌های مجازی‌سازی، طرح‌های آدرس‌دهی، ایزوله‌سازی عملکرد، قابلیت توسعه، تحمل خطا، نظارت، واسطها، امنیت، هزینه و مدیریت منابع به وجود آورده است. در این مقاله، ما بررسی‌ای روی تحقیقات اخیر بر روی شبکه‌های مرکز داده مجازی‌سازی شده انجام داده‌ایم. در ادامه ابتدا، خلاصه‌ایی از کارهای اخیر درباره مجازی‌سازی شبکه مرکز داده ارائه می‌دهیم، سپس این معماری‌ها را مقایسه خواهیم کرد. این کار اولین بررسی درباره تاریخچه مجازی‌سازی شبکه‌های مرکز داده است.

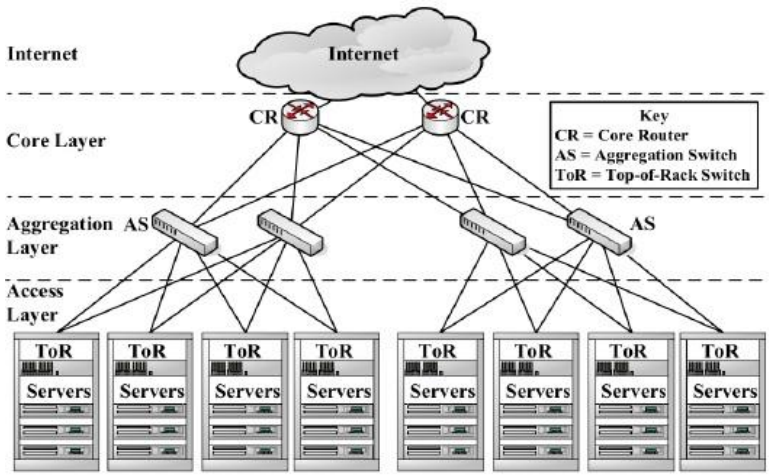
ادامه بررسی به صورت زیر سازماندهی شده است: بعد از معرفی تعاریف وابسته به مجازی‌سازی مرکز داده (بخش2)، پیشنهادات مرتبط به مجازی‌سازی شبکه مرکز داده را خلاصه می‌کنیم (بخش3) و آنها را از منظرهای مختلف مقایسه می‌کنیم (بخش4). و در نهایت مقاله خود را نتیجه گیری می‌کنیم (بخش5).

2-پیش زمینه

2-1- مرکز داده

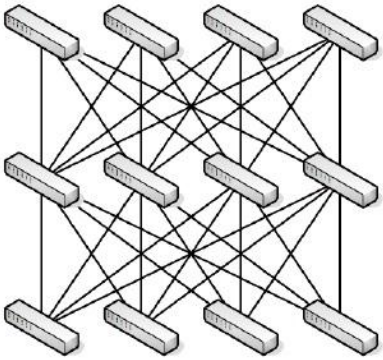
یک مرکز داده شامل سرورها، ذخیره‌سازها و دستگاه‌های شبکه (برای مثال، سوئیچ‌ها، روترها و کابل‌ها)، سیستم‌های توزیع برق و سیستم‌های خنک‌کننده است. و یک شبکه مرکز داده زیرساخت ارتباطی استفاده شده در مرکز داده است و توسط توپولوژی شبکه، تجهیزات روتینگ/سوئیچینگ و پروتکل استفاده شده توصیف می‌شود. در ادامه، توپولوژی رایج استفاده شده در مراکز داده و بعضی از سایر توپولوژی‌هایی که اخیراً پیشنهاد شده است، را ارائه می‌دهیم.

شکل 1 توپولوژی رایج شبکه مرکز داده را نشان می‌دهد[22]. در این توپولوژی، سوئیچ قرار گرفته در بالای قفسه در لایه دسترسی، اتصال به سرورهای داخل هر قفسه را ارائه می‌دهد. هر سوئیچ جمع‌آوری‌کننده در لایه جمع‌آوری ترافیک را از سوئیچ‌های لایه دسترسی چندگانه قرار گرفته در بالای قفسه به لایه هسته ارسال می‌کند. هر سوئیچ قرار گرفته در بالای قفسه به خاطر افزونگی به چندین سوئیچ جمع‌آوری کننده متصل است. لایه هسته اتصال امن بین سوئیچ‌های جمع‌آوری کننده و مسیریاب‌های هسته متصل به اینترنت را ارائه می‌دهد.



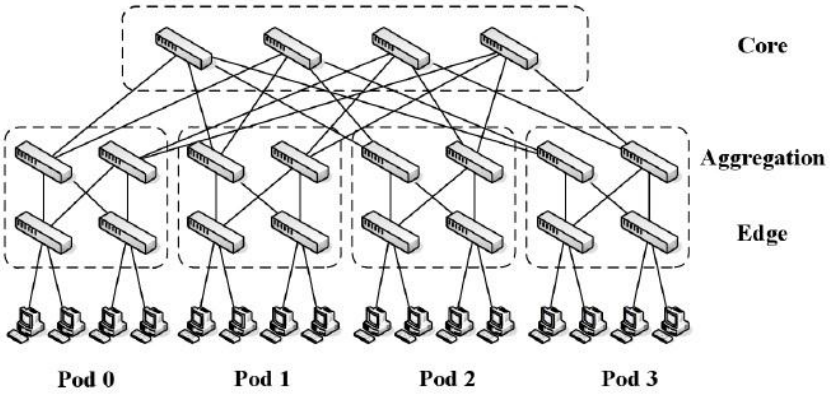
شکل 1: توپولوژی رایج شبکه مرکز داده [22]

توپولوژی Clos توپولوژی‌ای است که از چند مرحله از سوئیچ‌ها ساخته شده است[25]. هر سوئیچ در هر مرحله به تمام سوئیچ‌های مرحله بعد متصل می‌شود که تنوع گسترده‌ای از مسیر ارائه می‌دهد. شکل 2 مثالی از توپولوژی Clos سه مرحله‌ای را نشان می‌دهد.



شکل 2: توپولوژی Clos [25]

توپولوژی Fat-tree [5] نوع خاصی از توپولوژی Clos است که در ساختار درختی سازمان یافته است که در شکل 3 نشان داده شده است. توپولوژی از سوئیچ‌هایی با k پورت ساخته شده است که شامل k pod است: هر کدام از آنها دو لایه (جمع‌آوری و لبه) از k/2 سوئیچ‌ها را دارد. هر $(k/2)^2$ سوئیچ هسته یک پورت متصل به هر یک از k podها دارد. پورت iام از هر سوئیچ هسته به podiام متصل است بنابراین پورت‌های متوالی در لایه جمع‌آوری کننده هر سوئیچ pod به سوئیچ‌های هسته با k/2 گام متصل است. هر سوئیچ لبه مستقیماً به k/2 میزبان‌های نهایی متصل می‌شود؛ هر k/2 پورت‌های باقی مانده یک سوئیچ لبه به k/2 پورت‌های سوئیچ جمع‌آوری کننده متصل است [17].



شکل 3: توپولوژی Fat-tree (k=4) [25]

2-2- مجازی‌سازی مرکز داده

مرکز داده مجازی‌شده مرکز داده‌ایی است که همه یا بعضی از سخت‌افزارهایش (مثل سرورها، مسیریاب‌ها، سوئیچ‌ها و لینک‌ها) مجازی‌شده‌اند. معمولاً، سخت‌افزار فیزیکی با استفاده از نرمافزار یا سخت‌افزاری به نام hypervisor مجازی می‌شود که تجیهزات را به چند قسمت مستقل مجازی و ایزوله از هم تقسیم می‌کند.

یک مرکز داده مجازی یک مجموعه‌ای از منابع مجازی (ماشین‌های مجازی، سوئیچ‌های مجازی، مسیریاب‌های مجازی) است که با لینک‌های مجازی به هم مرتبط شده‌اند. در حالیکه مرکز داده مجازی‌شده یک مرکز داده فیزیکی با پیاده‌سازی تکنیک‌های مجازی‌سازی منابع است، یک مرکز داده مجازی نمونه‌ای منطقی از مرکز داده مجازی‌شده شامل زیرمجموعه‌ایی از

منابع مرکز داده فیزیکی است.

3- مروری بر پژوهش‌های انجام شده

مجازی‌سازی شبکه‌های مرکز داده هنوز در ابتدای راه است و تحقیقات اخیر عمدتاً بر روی چگونگی ارائه عملیات پایه و آتی شامل مشارکت منابع شبکه مرکز داده، طرح‌های ارسال بسته، ایزوله کردن عملکرد شبکه و تحمل خطا متمرکز است. برهمین اساس، ما به چهار موضوع در این بررسی توجه کرده‌ایم:

- طرح‌های ارسال بسته با مشخص کردن قوانین استفاده شده برای ارسال بسته‌ها بین گره‌های مجازی
- تضمین پهنای باند و مکانیزم‌های به اشتراک‌گذاری پهنای باند مربوطه که ایزوله کردن عملکرد شبکه و اشتراک کاراتر منابع شبکه را ارائه می‌دهد.
- تکنیک‌های چندمسیری استفاده شده برای پخش ترافیک بین مسیرهای مختلف برای بهبود متعادل‌سازی بار
- تحمل خطا که یک نیازمندی مهم در سرویس‌های پایه مرکز داده، برای افزایش قابلیت اطمینان و دسترس‌پذیری است.

علی‌رغم اینها، ویژگی‌های با ارزش دیگری مثل امنیت، قابلیت برنامه‌ریزی، قابلیت مدیریت و مصرف کمتر انرژی وجود دارد که وقتی مراکز داده را مجازی می‌کنیم در نظر می‌گیریم.

در جدول 1 دسته‌بندی‌ای از پروژه‌های بررسی شده را براساس ویژگی‌هایی که پوشش می‌دهند ارائه می‌دهیم و تاکید می‌کنیم که پروژه ممکن است بیشتر از یک ویژگی را بررسی کند. چک‌مارک ویژگی‌های اصلی پیشنهادات بررسی شده را نشان می‌دهد.

جدول 1: دسته‌بندی پیشنهادهای بررسی شده

پیشنهاد	ویژگی				
	طرح ارسال	تضمین پهنای باند	چند مسیری	به اشتراک‌گذاری پهنای باند	تحمل خطا
مرکز داده سنتی	✓		✓		
SPAIN			✓		
Diverter	✓				
NetLord	✓		✓		
VICTOR	✓				
VL2	✓		✓		
PortLand	✓		✓		
Oktopus		✓			
SecondNet	✓	✓			
Seawall				✓	
Gatekeeper		✓			
NetShare			✓	✓	
SEC2	✓				
CloudNaaS	✓	✓			
Zeppelin	✓				✓
تحمل خطا در معماری مجازی‌سازی شبکه مرکز داده			✓		✓
اختصاص شبکه مجازی برای تحمل خطا با کارایی پهنای باند				✓	✓

3-1- مرکز داده سنتی

مجازی‌سازی در معماری‌های فعلی مرکز داده عموماً توسط مجازی‌سازی سرور به دست می‌آید. هر مستاجر صاحب یک گروه از سرورهای مجازی است و ایزوله‌سازی بین مستاجران از طریق شبکه‌های محلی مجازی به دست می‌آید. مراکز داده مرتبط به این طراحی ساده می‌توانند با استفاده سوئیچ‌های مناسب و فناوری‌های hypervisor رایج اجرا شود. در کنار این، مستاجران می‌توانند فضاهای آدرس‌دهی لایه دو و لایه سه خود را تعریف کنند.

محدودیت اصلی معماری‌های فعلی مرکز داده قابلیت توسعه است زیرا سوئیچ‌های مناسب برای مدیریت تعداد زیادی ماشین مجازی و در نتیجه میزان ترافیک طراحی نشده است. علاوه بر این، چون شبکه‌های محلی مجازی برای دستیابی به ایزوله‌سازی بین مستاجران استفاده می‌شوند، تعداد مستاجران محدود به 4096 است (تعداد شبکه‌های محلی مجازی طبق استاندارد 802.1q).

3-2- Diverter

پشتیبانی از تقسیم‌بندی منطقی شبکه‌های IP برای هم‌اقامتی بهتر برنامه‌های کاربردی و سرویس‌های مورد نیاز در محیط‌های چند مستاجره با مقیاس بزرگ مثل مراکز داده ضروری است. Diverter [1] رویکردی صرفاً نرم‌افزاری برای مجازی‌سازی شبکه مرکز داده است که فرض می‌کند به پیکربندی سوئیچ‌ها یا روترها نیاز ندارد.

Diverter در یک ماژول نرم‌افزاری به نام VNET که در هر ماشین فیزیکی نصب شده است، اجرا می‌شود. وقتی که ماشین مجازی یک فریم اترنت ارسال می‌کند، VNET آدرس‌های فیزیکی مبدأ و مقصد را به ترتیب با یکی از ماشین‌های فیزیکی که ماشین‌های مجازی مبدأ و مقصد را میزبانی می‌کند، جایگزین می‌کند. سپس سوئیچ‌ها ارسال بسته را با استفاده از آدرس‌های فیزیکی ماشین‌های فیزیکی انجام می‌دهند. Diverter نیاز دارد که هر ماشین مجازی یک فرمت آدرس IP برای کد کردن شناسه مستاجر، زیرشبکه و آدرس ماشین مجازی داشته باشد بنابراین هیچ تصادم آدرسی بین مستاجران اتفاق نمی افتد به صورت خلاصه، Diverter مجازی‌سازی لایه سه شبکه را ارائه می‌دهد که به هر مستاجر اجازه کنترل زیرشبکه‌های IP و آدرس‌های ماشین‌های مجازی خودشان را می‌دهد. محدودیت اصلی این پیشنهاد عدم ارائه تضمین کیفیت سرویس است، که نویسنده‌ها آن را به عنوان کارهای آتی در نظر گرفته‌اند.

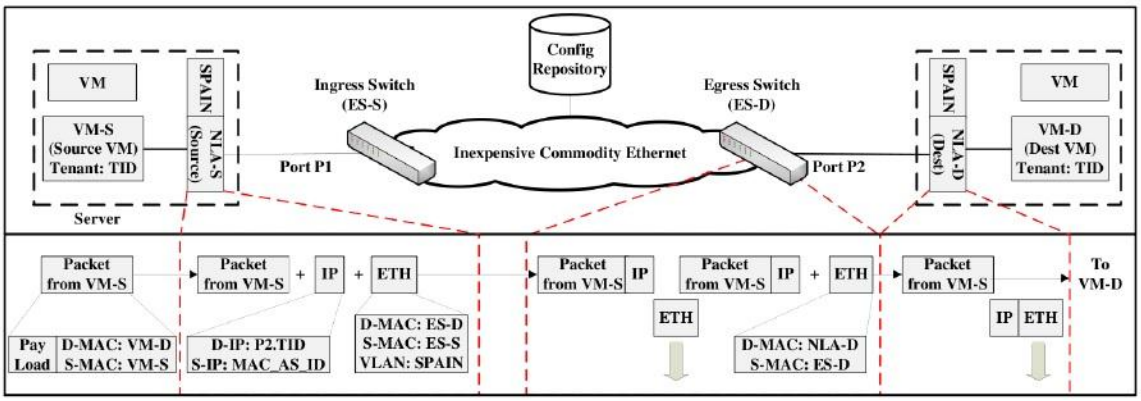
3-3- NetLord

NetLord [15] معماری شبکه‌ای است که برای قابلیت توسعه تعداد مستاجران در مراکز داده تلاش می‌کند. معماری، فضای آدرس لایه دو و لایه سه مستاجر را مجازی می‌کند که به مستاجران اجازه طراحی و پیاده‌سازی فضای آدرس‌دهی خود را براساس نیازشان و پیاده‌سازی برنامه‌های کاربردی می‌دهد.

ایده کلیدی NetLord در شکل 4 ارائه شده است عامل NetLord مستقر بر روی هر سرور فیزیکی، که تمام ماشین‌های مجازی روی سرور را کنترل می‌کند. یک بسته لایه دو را با افزودن یک هدر لایه دو و یک هدر لایه سه اضافی کپسوله می‌کند. آدرس‌های فیزیکی لایه دو اضافی آدرس‌های سوئیچ‌های مبدأ و مقصد را که به ترتیب سوئیچ‌های متصل به سرور میزبان ماشین‌های مجازی مبدأ و مقصد هستند را تعیین می‌کند. و آدرس‌های IP مبدأ و مقصد لایه سه اضافی به ترتیب فضای آدرس فیزیکی مستاجر مبدأ و پورت ورودی سوئیچ و شناسه مستاجر ماشین مجازی مقصد را تعیین می‌کند.

بسته بر روی شبکه مرکز داده بر روی مسیر انتخاب شده توسط الگوریتم انتخاب شبکه محلی مجازی و SPAIN [16] منتقل می‌شود. یک عامل NetLord بسته‌ها را به سرور مقصد به ماشین مجازی مقصد، با استفاده از شناسه‌های مستاجر و فضای آدرس فیزیکی و آدرس مقصد ماشین مجازی در بسته کپسوله مستاجر، ارسال می‌کند و بسته به مقصد می‌رسد. گرچه

این معماری، ایزوله کردن بین مستاجران را ارائه می‌دهد، از هیچ تضمین پهنای باندی استفاده نمی‌کند.

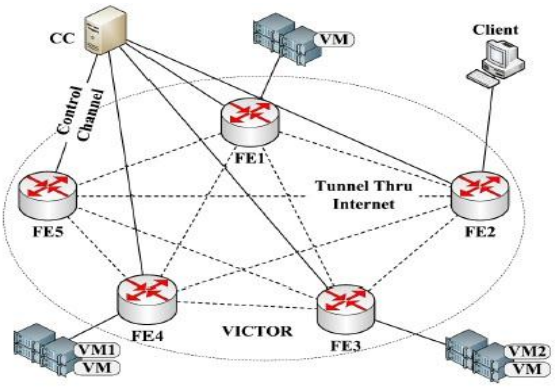


شکل 4: معماری NetLord [15]

VICTOR -3-4

مستاجران ابر نیاز به مهاجرت سرویس‌ها بین مراکز داده دارند تا بار را در داخل و بین مراکز داده متعادل کنند و یا عملکرد سرویس‌هایشان را بهینه کنند. از طرف دیگر، کاربران ابر می‌خواهند تحویل سریع و کارا از سرویس‌ها و داده داشته باشند. یک رویکرد که امکان دستیابی به اهداف بالا از منظر مستاجران و کاربران را می‌دهد مهاجرت ماشین‌های مجازی است. برای اجتناب از وقفه سرویس، یک ماشین مجازی باید یک آدرس IP مشابه را در طول مهاجرت حفظ کند. گرچه چالشی برای مهاجرت در شبکه IP مشابه وجود ندارد، ارائه مهاجرت بر روی شبکه‌های مختلف آسان نیست. VICTOR [9] معماری شبکه‌ای برای پشتیبانی از مهاجرت ماشین‌های مجازی بین شبکه‌های مختلف است که امکان حفظ آدرس‌های IP اصلی آنها را فراهم می‌کند.

ایده اصلی VICTOR که در شکل 5 نشان داده شده است ایجاد یک خوشه عنصر ارسال‌کننده است. یک کنترل‌کننده مرکزی مجموعه‌ای از عناصر ارسال‌کننده را کنترل می‌کند که بر روی چندین شبکه توزیع شده‌اند و از مهاجرت ماشین‌های مجازی بین شبکه‌های مختلف پشتیبانی میکنند. این کنترل‌کننده مرکزی دارای جداولی برای توپولوژی بین عناصر ارسال‌کننده و اتصال بین هر ماشین مجازی و عنصر ارسال‌کننده است و بوسیله آنها مسیر را برای ارسال بسته‌ها مسیریابی می‌کند. محدودیت اصلی VICTOR نیاز به پشتیبانی از ارسال‌کننده‌های بلوک اطلاعات با سایز بزرگ است که به موضوع قابلیت توسعه عناصر ارسال‌کننده مربوط می‌شود.



شکل 5: معماری VICTOR [9]

VL2 -3-5

VL2 [2] معماری شبکه مرکز داده‌ای است که هدفش دستیابی به انعطاف‌پذیری در تخصیص منابع است. در VL2 تمام سرورهای متعلق به مستاجران یک فضای آدرس‌دهی را بدون توجه به مکان فیزیکی آنها به اشتراک می‌گذارند به این معنی که هر سروری می‌تواند به هر مستاجری اختصاص داده شود.

VL2 براساس توپولوژی Clos است. بسته‌ها با استفاده از دو نوع آدرس IP ارسال می‌شوند: آدرس‌های براساس مکان و آدرس‌های براساس برنامه کاربردی که توسط سوئیچ‌ها و سرورها استفاده می‌شوند. سوئیچ‌ها بسته‌ها را فقط با استفاده از آدرس‌های براساس مکان ارسال می‌کنند. بسته در مبدأ با آدرس بر اساس مکان کپسوله شده و ارسال می‌شود. در سوئیچ مقصد این آدرس به آدرس بر اساس برنامه کاربردی برگردانده شده و به سرور مورد نظر تحویل داده می‌شود.

گسترش بین فضاهای آدرس‌دهی سوئیچ‌ها و سرورها، قابلیت توسعه VL2 را بهبود می‌بخشد زیرا سوئیچ‌های قرار گرفته در بالای قفسه اطلاعات ارسال شده برای تعداد زیادی سرور را نگهداری نمی‌کنند. یکی از محدودیت‌های VL2 کمبود تضمین مطلق پهنای باند بین سرورها است که مورد نیاز تعداد زیادی از برنامه‌های کاربردی است.

PortLand -3-6

قابلیت توسعه تعداد ماشین‌های مجازی، مهاجرت کارای ماشین مجازی و مدیریت ساده ویژگی‌های مهم مراکز داده فعلی و نسل بعدی هستند. PortLand [20] تمام این موضوعات را برای توپولوژی چندریشه‌ای Fat-tree بررسی کرده است (شکل 3).

ایده اصلی PortLand استفاده از آدرس‌های شبه فیزیکی سلسله مراتبی ماشین‌های مجازی برای مسیریابی لایه دو است. یک سوئیچ لبه، که یک سرور میزبان یک ماشین مجازی به آن متصل است، یک آدرس فیزیکی واقعی از ماشین مجازی را به آدرس شبه فیزیکی نگاشت می‌کند. مدیریت محصول (فرایندی که بر روی یک ماشین اختصاصی اجرا می‌شود) مسئول کمک به پاسخگویی ARP، چندپخشی و تحمل شکست است. موقعیت سوئیچ در توپولوژی ممکن است به صورت دستی توسط مدیر یا به صورت اتوماتیک از طریق پروتکل کشف مکان پیشنهادی توسط نویسندگان تنظیم شود که به ویژگی‌های توپولوژی لایه زیرین وابسته است.

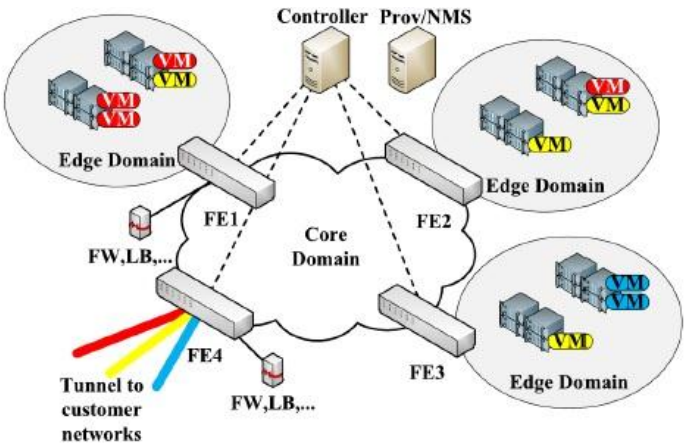
موضوعاتی که این معماری را محدود می‌کنند، نیاز به توپولوژی Fat-tree چند ریشه‌ایی، و آسیب‌پذیری معماری در برابر حملات مخرب بر روی سرور پاسخگویی درخواست‌های ARP است که می‌تواند به عدم دسترس‌پذیری سرویس منجر شود و نیز اینکه هر سوئیچ لبه باید حداقل نیمی از پورت‌هایش به سرورها متصل باشد.

SEC2 -3-7

یکی از موضوعات مهم امنیتی ایزوله کردن شبکه‌های مجازی اختصاص داده شده به مستاجران مختلف است. SEC2 [10] معماری شبکه مرکز داده است که از تکنیک‌های مجازی‌سازی شبکه برای ارائه سرویس امنیت ارتجاعی رایانش ابری که در شکل 6 نشان داده شده است، استفاده می‌کند. مجازی‌سازی شبکه از طریق عناصر ارسال‌کننده و کنترل‌کننده‌های مرکزی پیشتیبانی می‌شود. عناصر ارسال‌کننده ضرورتاً سوئیچ‌های اترنت با قابلیت کنترل از راه دور کنترل‌کننده‌های مرکزی هستند که نگاشت آدرس‌ها و پایگاه داده‌های سیاست ها را ذخیره می‌کنند. معماری شبکه دو سطح دارد: دامنه هسته و دامنه‌های لبه چندگانه که با میزبان‌های فیزیکی همکاری می‌کند. برای ایزوله کردن مشتریان متفاوت با هر دامنه لبه، SEC2 از شبکه محلی مجازی با محدوده محدود شده با دامنه لبه مشابه استفاده می‌کند، اگر مشتری بخواهد به یک ماشین مجازی دسترسی عمومی داشته باشد، عنصر ارسال‌کننده تمام بسته‌های خارجی را به عبور از فایروال و جعبه‌های میانی ترجمه آدرس شبکه قبل از

رسیدن به شبکه‌های خصوصی مجبور می‌کند. مزیت SEC2 عدم نیاز به روترها یا سوئیچ‌های مشخص شده از طریق ورودی شبکه مرکز داده است. علاوه بر این، SEC2 از مهاجرت ماشین مجازی [9] و سرویس ابر مجازی خصوصی پشتیبانی می‌کند که هر شبکه خصوصی کاربر در ابر به شبکه سایت با استفاده از شبکه خصوصی مجازی متصل است[8].

یکی از محدودیت‌های SEC2 این است که یک دامنه لبه نمی‌تواند از شبکه‌های محلی مجازی با بیش از 4K مستاجر مختلف پشتیبانی کند. علاوه بر این، از آنجایی که عناصر ارسال‌کننده هدر آدرس فیزیکی بیرونی را وقتی که ماشین مجازی مقصد در دامنه لبه نیست، اضافه می‌کنند، SEC2 نیاز به سوئیچ‌هایی دارد که از فریم‌های بزرگ پشتیبانی کنند.



شکل 6: معماری SEC2 [10]

SPAIN -3-8

پروتکل درخت پوشای فعلی مورد استفاده برای شبکه‌های محلی اترنت بزرگ برای پشتیبانی از شبکه‌های مرکز داده مدرن کافی نیست. SPAIN [16] (واگذاری راه هوشمند در شبکه) از شبکه محلی مجازی پشتیبانی شده توسط سوئیچ‌های اترنت موجود استفاده می‌کند تا توپولوژی‌های چندمسیری اختیاری را ارائه دهد.

SPAIN مسیرهای منقطع بین جفت سوئیچ‌های لبه را محاسبه کرده و شبکه‌های محلی مجازی را برای شناسایی این مسیرها از قبل تنظیم می‌کند. عامل میزبان نهایی نصب شده بر روی هر میزبان جریان‌ها را از طریق مسیرهای شبکه‌های محلی مجازی مختلف گسترش می‌دهد. برای بهبود توازن بار و اجتناب از شکست، عامل می‌تواند مسیرها را برای بعضی از جریان‌ها تغییر دهد. عامل همچنین مسیرهای شکست را نیز تشخیص داده و بسته‌ها را اطراف شکست با استفاده از مسیرهای متفاوت دوباره مسیریابی می‌کند.

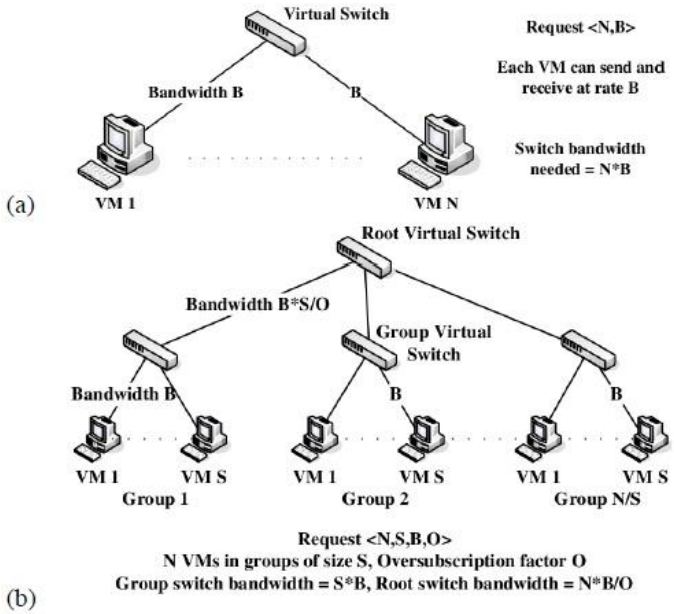
درحالی که SPAIN چندمسیری را ارائه می‌دهد و تعادل بار و تحمل شکست را بهبود می‌بخشد، این پیشنهاد بعضی از موضوعات قابلیت توسعه را در بردارد. SPAIN نیاز دارد تا سوئیچ‌ها ورودی‌های چندگانه را برای هر مقصد و شبکه محلی مجازی حفظ کنند. این فشار بیشتری بر روی جدول ارسال سوئیچ نسبت به استاندارد اترنت ایجاد می‌کند. بنابراین، با استفاده از استاندارد 802.1q تعداد مسیرها به تعداد شبکه‌های محلی مجازی محدود می‌شود.

Oktopus -3-9

گرچه ارائه دهندگان زیرساخت منابع محاسباتی مورد تقاضای مستاجر را از طریق تخصیص ماشین‌های مجازی در مراکز داده ارائه می‌دهند، آنها از تضمین عملکرد بر روی منابع شبکه به مستاجران، پشتیبانی نمی‌کنند. Oktopus [11] اجرای دو

انتزاع شبکه مجازی (خوشه مجازی و خوشه oversubscribed مجازی) برای کنترل تعادل بین تضمین عملکرد پیشنهادی به مستاجران و هزینه‌های آنها و درآمد ارائه دهندگان می‌باشد. Oktopus فقط عملکرد برنامه کاربردی را افزایش نمی‌دهد بلکه انعطاف‌پذیری بیشتری به ارائه دهندگان زیرساخت ارائه داده و به مستاجران امکان یافتن تعادل بین عملکرد بالاتر برنامه کاربردی و هزینه کمتر را می‌دهد.

خوشه مجازی نشان داده شده در شکل 7-a تصویری از داشتن تمام ماشین‌های مجازی متصل به یک سوئیچ مجازی بدون oversubscribed ارائه می‌دهد. یک خوشه oversubscribed مجازی که در شکل 7-b شرح داده شده است، از خوشه دولایه oversubscribed تقلید می‌کند که مجموعه‌ای از کلاسترهای مجازی است که به صورت داخلی از طریق سوئیچ ریشه مجازی متصل شده است که مناسب ویژگی‌های برنامه‌های کاربردی الگوهای ارتباطی محلی است. Oktopus از الگوریتم حریصانه برای تخصیص منابع به مرکز داده مجازی استفاده می‌کند. محدودیت اصلی Oktopus این است که فقط با توپولوژی‌های درختی شبکه فیزیکی کار می‌کند. بنابراین یک سوال مطرح این است که چگونه می‌شود انتزاع Oktopus را برای سایر توپولوژی‌ها اجرا کرد.



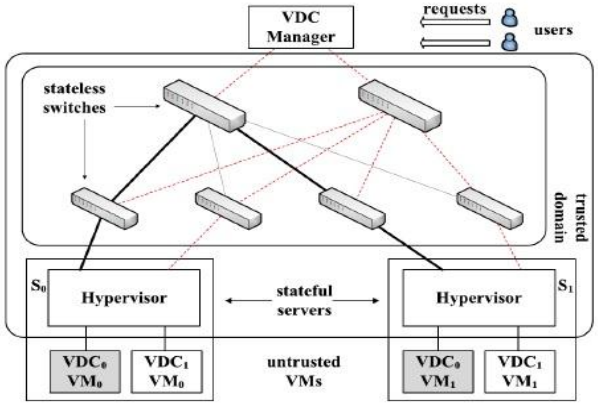
شکل 7: انتزاعی از Oktopus. a) خوشه مجازی، b) خوشه oversubscribed مجازی [11]

SecondNet -3-10

SecondNet [4] روی ارائه تضمین پهنای باند بین ماشین‌های مجازی چندگانه در مرکز داده مجازی‌شده با چندین مستاجر تمرکز کرده است. علاوه بر محاسبات و ذخیره‌سازی، این معماری نیازهای پهنای باند را وقتی که یک مرکز داده مجازی مستقر می‌شود در نظر گرفته است.

مولفه اصلی معماری SecondNet که در شکل 8 نشان داده شده است، مدیر مرکز داده مجازی است که مراکز داده مجازی را براساس ماتریس نیازمندیها که پهنای باند درخواستی بین جفت های ماشین مجازی را تعریف می‌کند، ایجاد می‌کند. SecondNet سه نوع سرویس اصلی را تعریف می‌کند: سرویس تضمینی انتها به انتهای اولویت بالا (نوع 0)، بهتر از سرویس کمترین تلاش (نوع 1) که تضمین پهنای باند را برای گام‌های اول و آخر مسیر ارائه می‌دهد. و سرویس کمترین تلاش (نوع 2).

SecondNet از طرح ارسال اصلاح شده به نام مسیریابی مبدأ سوئیچینگ پورت استفاده می‌کند که بسته‌ها را با استفاده از شماره پورت‌های از پیش تعریف شده به جای آدرس‌های فیزیکی ارسال می‌کند. در این روش، سوئیچ‌های واسط هیچ تصمیمی درباره ارسال نمی‌گیرند.



شکل 8: معماری SecondNet [4]

SecondNet با انتقال اطلاعات مربوط به رزرو پهنای باند از سوئیچ‌ها به hypervisors سرور به قابلیت توسعه بالایی دست می‌یابد. در کنار اینها، SecondNet به منابع اجازه می‌دهد به صورت پویا به مراکز داده مجازی اضافه شده یا از مراکز داده مجازی حذف شوند. با استفاده از مهاجرت، SecondNet همچنین قادر به مدیریت شکست‌ها و کاهش تکه تکه کردن منابع است. علاوه بر این، مسیریابی مبدأ سوئیچینگ پورت می‌تواند با MPLS [7] اجرا شود که پیاده‌سازی را ساده می‌کند. محدودیت اصلی SecondNet این است که عملکردش به توپولوژی فیزیکی شبکه وابسته است.

Gatekeeper -3-11

Gatekeeper [12] روی ارائه پهنای باند تضمینی بین ماشین‌های مجازی در مرکز داده با چندین مستاجر تمرکز می‌کند و به بهره‌برداری بالایی از پهنای باند دست می‌یابد. Gatekeeper به این موضوع با تعریف کمترین نرخ تضمین و بیشترین نرخ مجاز برای هر جفت ماشین مجازی می‌پردازد. این پارامترها می‌تواند برای دستیابی کمترین تضمین پهنای باند پیکربندی شود در حالیکه که تضمین می‌کند ظرفیت‌های لینک به صورت کارا توسط مستاجران استفاده شود. Gatekeeper یک یا چند سوئیچ منطقی ایجاد می‌کند که ماشین‌های مجازی متعلق به یک مستاجر را بهم متصل می‌کند. کارت شبکه مجازی هر ماشین مجازی دریافتی نرخ ترافیک ورودی را با استفاده از مجموعه‌ای از شمارنده‌ها مانیتور می‌کند و ازدحام را به کارت شبکه مجازی فرستنده گزارش می‌کند که تضمین کمینه را با بیشترین مقدار افزایش می‌دهد. محدود کننده نرخ در فرستنده از این اطلاعات برای کنترل نرخ ترافیک استفاده می‌کند تا سطح ازدحام را کاهش دهد.

مانند بسیاری از طرح‌های موجود، Gatekeeper سایر معیارهای عملکرد مثل تاخیر را در نظر نمی‌گیرد. در کنار این، Gatekeeper هنوز تحت توسعه است: ویژگی‌های کلیدی مثل ایجاد پویایی و تشخیص محدود کننده‌های نرخ هنوز اجرا نشده‌اند.

CloudNaaS -3-12

CloudNaaS [23] معماری شبکه مجازی است که پشتیبانی کارا برای استقرار و مدیریت برنامه‌های کاربردی تجاری در

ابرها را ارائه می‌دهد. هدف CloudNaaS ارائه چارچوب جامع و وسیع برای اجرای برنامه‌های کاربردی تجاری در ابرهاست.

CloudNaaS به ارسال OpenFlow برای دست‌یابی به اهداف متکی است. استقرار برنامه کاربردی در CloudNaaS شامل چندین گام است. ابتدا کاربر نهایی نیازهای شبکه برای کنترل‌کننده ابر را با استفاده از اولیه های تعریف شده توسط زبان سیاست شبکه را مشخص می‌کند. پس از اینکه نیازهای شبکه به ماتریس ارتباطی ترجمه شد، کنترل‌کننده ابر مکان ماشین‌های مجازی و قوانین تضمین سطح شبکه را تعیین می‌کند که می‌تواند روی سوئیچ‌ها نصب شود. در حال حاضر، CloudNaaS از صندوق بسته‌بندی اکتشافی حریصانه اصلاح شده، برای قرار دادن ماشین‌های مجازی استفاده می‌کند که ملاحظات محل ارتباطات را در نظر می‌گیرد. علاوه بر این، CloudNaaS تکنیک‌هایی را ارائه می‌دهد تا تعداد ورودی‌های مورد نیاز برای هر سوئیچ را کاهش دهد. CloudNaaS همچنین از مکانیزم‌های آنلاین برای مدیریت شکست ها و تغییرات در مشخصات سیاست شبکه با استفاده از تامین مجدد مراکز داده مجازی پشتیبانی می‌کند.

یکی از محدودیت‌های CloudNaaS محدود کردن ترافیک به مسیرهای کمی است که ممکن است به ازدحام یا بهره‌برداری ضعیف شبکه منجر شود. پیدا کردن توازن بهتر بین قابلیت توسعه و بهره‌برداری از شبکه هنوز مشکل چالش برانگیزی برای CloudNaaS است.

Seawall -3-13

Seawall [3] طرح تخصیص پهنای باند است که امکان تعریف نحوه اشتراک پهنای باند در شبکه مرکز داده با چندین مستاجر را به ارائه دهندگان زیرساخت می‌دهد. ایده Seawall اختصاص وزن به عناصر تولید کننده ترافیک در شبکه و تخصیص پهنای باند براساس این وزن‌ها به روش مناسب است. Seawall از تونل های کنترل ازدحام بین جفت عناصر شبکه استفاده می‌کند تا سیاست‌های اشتراک پهنای باند را اجرا کند.

Seawall ایزوله‌سازی پهنای باند بین مستاجران مختلف را ممکن ساخته و از مستاجران مخرب در مصرف تمام منابع شبکه جلوگیری می‌کند. در کنار این، Seawall نیاز دارد که ماشین فیزیکی اطلاعات وضعیت را فقط برای صاحبان عناصر حفظ کند که این قابلیت توسعه را بهبود می‌بخشد. علاوه بر این، Seawall امکان وزن‌گذاری قابل اصلاح به صورت پویا برای تطبیق تغییرات در نیازهای مستاجران را می‌دهد. گرچه Seawall شکست‌ها را به صراحت بررسی نمی‌کند، اما با شرایط پویای شبکه سازگار است و تحمل خطا را دارد.

نمونه اولیه Seawall فقط روی ویندوز 7 و Hyper-v پیاده‌سازی شده است، علاوه بر این بدون کنترل پذیرش، بعید است که Seawall بتواند در هنگام افزایش تعداد عناصر، به تضمین کامل پهنای باند برسد.

NetShare -3-14

NetShare [24] مسئله تخصیص پهنای باند در شبکه‌های مرکز داده مجازی‌شده را به عهده گرفته و مکانیزم مالتی پلکس کردن آماری را پیشنهاد می‌دهد که به هیچ تغییری در سوئیچ‌ها یا روترها احتیاج ندارد. NetShare پهنای باند را برای مستاجران به روش مناسبی تخصیص داده و به بهره‌برداری بالایی از لینک برای ارائه دهندگان زیرساخت دست یافته است. در این روش، یک سرویس/ برنامه کاربردی/گروه نمی‌تواند پهنای باند در دسترس لینک‌های به اشتراک گذاشته شده را با باز کردن اتصالات بیشتر به زور به دست آورد.

NetShare می‌تواند به سه روش ممکن اجرا شود: تخصیص گروهی، کنترل (گلوگاه) نرخ و تخصیص مرکزی. NetShare از تخصیص گروهی برای کنترل جریان‌های TCP استفاده می‌کند. تخصیص گروهی از صف‌بندی عادلانه برای ارائه تخصیص پهنای باند عادلانه بین سرویس‌های مختلف استفاده می‌کند و از طریق چرخش عادلانه اجرا می‌شود[19]. کنترل نرخ برای کنترل ترافیک تولید شده توسط منابع UDP و اجتناب از مصرف پهنای باند بیش از اندازه استفاده می‌شود. و برای اجرای

سیاست‌های عمومی مثل تخصیص پهنای باند استفاده نشده به جریان‌های خاص، این طرح از تخصیص پهنای باند مرکزی استفاده می‌کند. NetShare به پروتکل مسیریابی برای مدیریت شکست‌ها متکی است و چندمسیری با استفاده از چند مسیری هزینه برابر ممکن است.

قابلیت توسعه NetShare می‌تواند یک موضوع باشد زیرا صف‌ها در هر پورت سوئیچ برای هر برنامه کاربردی / سرویس پیکربندی می‌شود. در ضمن، هدف NetShare دستیابی به تخصیص عادلانه پهنای باند است بنابراین هیچ تضمین کاملی از پهنای باند را به سرویس‌ها ارائه نمی‌دهد.

Zeppelin -3-15

Zeppelin [14] از چندین لایه از برچسب‌ها برای رسیدن به ایزوله‌سازی و مسیرهای طراحی شده در شبکه مرکز داده استفاده می‌کند. برچسب گذاری بسته با استفاده از MPLS با یک پنل کنترل نرم‌افزار تعریف ساختار شبکه متمرکز شده، که کنترل OpenFlow از سوئیچ‌های مرکز داده را اجرا می‌کند. Zeppelin هیچ فرضی درباره طراحی شبکه ندارد. تنها ملزومات Zeppelin که باید در ساختار شبکه وجود داشته باشد، این است که سوئیچ‌های قرارگرفته در بالای قفسه در یک ساختار مش کامل به هم متصل باشند، و بتوانند بسته‌های برچسب گذاری شده MPLS ارسالی را در تونل overlay به سمت مقصد و بسته‌های برچسب گذاری شده MPLS دریافتی را به سوئیچ مجازی و سرور مشخص شده در برچسب مسیریابی کنند.

علاوه بر این سوئیچ قرار گرفته در بالای قفسه مبدأ ممکن است چندین تونل overlay در دسترس برای یک سوئیچ قرار گرفته در بالای قفسه مقصد مشخص داشته باشد، که اجازه انجام مسیریابی چند مسیری را می‌دهد.

رویکرد Zeppelin بر اساس یک پنل کنترل OpenFlow و یک پنل داده MPLS است. Zeppelin از دولایه از برچسب‌های MPLS استفاده می‌کند: یکی مشخص کننده شبکه مجازی که مستاجر در آن قرار گرفته و دیگری مشخص کننده مسیر مسیریابی در میان شبکه. لایه مسیر مسیریابی خودش شامل دو برچسب است: یکی مشخص کننده لینک بین مقصد آخرین گام سوئیچ مجازی و آخرین گام سوئیچ قرار گرفته در بالای قفسه، و دیگری مشخص کننده مسیر در ستون فقرات شبکه. متضاد با استفاده از MPLS در شبکه‌های گسترده، جایی که برچسب‌های MPLS خارج از لینک و مسیریابی که روی آن اختصاص داده شده‌اند، بی‌معنی هستند، در Zeppelin برچسب‌های MPLS در سراسر مرکز داده ارتباط معنایی دارند. کنترل‌کننده مرکزی تخصیص برچسب‌های MPLS را مدیریت کرده و سوئیچ‌های مجازی و سوئیچ‌های قرار گرفته در بالای قفسه را با استفاده از پروتکل OpenFlow برنامه‌ریزی می‌کند.

3-16- تحمل خطا در معماری‌های مجازی‌سازی شبکه مرکز داده

تحمل خطا یک نیازمندی مهم در سرویس‌های پایه مرکز داده، برای افزایش قابلیت اطمینان و دسترس‌پذیری است. شکست در مرکز داده را می‌توان به شکست در لینک و شکست در سرور تقسیم‌بندی کرد. در این کار [21] یک مکانیزم تحمل خطا برای مدیریت شکست‌های سرور توسط مهاجرت کارای ماشین‌های مجازی میزبانی شده در سرور شکست خورده به یک مکان جدید ارائه شده است. این مکانیزم به این صورت عمل می‌کند که پس از رخداد یک شکست، درخواست‌هایی را که مسیرشان از سرور شکست خورده عبور می‌کند را پیدا کرده و مسیر جدیدی را برای مسیر قطع شده به دلیل شکست مسیریابی می‌کند، و سپس ماشین‌های مجازی میزبانی شده در سرور شکست خورده را مکان‌دهی مجدد می‌کند. به این ترتیب می‌تواند خطاها را در مرکز داده برطرف کند.

با اجرای طرح تحمل خطا مراکز داده مجازی زیادی تحت تأثیر خرابی یک سرور خاص قرار می‌گیرند، برای کاهش تأثیر شکست‌های سرور روی مراکز داده مجازی میزبانی شده در مرکز داده نیز یک رویکرد تعادل بار بر اساس خوشه‌بندی، با استفاده از هیوریستیک‌ها، ارائه شده که مراکز داده مجازی را به صورت کارا در مرکز داده اختصاص می‌دهد. این

هیوریستیک‌های اعمال شده نیاز به اطلاعات اضافه‌تری درباره دسترس‌پذیری منابع در هر خوشه دارد. که گرچه این موضوع کارایی الگوریتم اختصاص درخواست‌های مرکز داده مجازی را کاهش می‌دهد، ولی به قابلیت اطمینان بیشتری دست پیدا می‌کند. این رویکرد تلاش می‌کند درخواست‌های مرکز داده مجازی را به صورت مساوی در سراسر شبکه مرکز داده اختصاص دهد. این طرح در شبکه‌های BCube و DCell نتایج خوبی ارائه داده است و می‌تواند روی سایر توپولوژی‌های شبکه مانند Fat-Tree نیز مورد استفاده قرار بگیرد.

17-3- اختصاص شبکه مجازی برای تحمل خطا با کارایی پهنای باند

در این کار [26] روی ویژگی‌های کارایی به اشتراک‌گذاری منابع شبکه فیزیکی توسط شبکه‌های مجازی همزمان تمرکز شده است، چگونگی اختصاص منابع فیزیکی به گره‌ها و لینک‌های شبکه مجازی که روی تحمل خطا در شبکه مجازی تأثیرگذار خواهد بود تصریح و یک مدل اختصاص شبکه مجازی برای مراکز داده چند مستاجره ارائه شده است.

در مدل چند مستاجره هر گره و لینک هر شبکه مجازی به یک سرور و مسیر شبکه فیزیکی نگاشت می‌شود. در این نوع شبکه مرکز داده وقتی که یک شکست در سوئیچ یا لینک فیزیکی اتفاق می‌افتد و لینک‌های شبکه مجازی نگاشت شده به آن را مختل می‌کند، بازیابی سوئیچ یا لینک فیزیکی موجب بازیابی لینک‌های شبکه مجازی می‌شود، شبکه مجازی به تنهایی قابلیت بازیابی خطا را ندارد. اما اگر شکست در سرور فیزیکی رخ دهد و روی شبکه مجازی‌ای که ماشین‌های مجازی آن روی این سرور قرار گرفته‌اند تأثیر بگذارد، شبکه مجازی باید با استفاده از مکانیزم بازیابی خطای خودش ماشین‌های مجازی را بازیابی کند. به این صورت که هر یک از ماشین‌های مجازی در یک شبکه مجازی با یک ماشین مجازی آماده به کار داغ جفت است و یک ماشین مجازی آماده به کار سرد در هر سرور فیزیکی وجود دارد، تا در صورت بروز خطا جایگزین آن شود.

در این کار هدف اختصاص شبکه مجازی، کمینه کردن پهنای باند از دست رفته است هنگامی که یک شکست در شبکه فیزیکی رخ می‌دهد. نتایج ارزیابی های انجام شده نشان داده است که تعادل بین پهنای باند از دست رفته و پهنای باند مورد نیاز بین سرورهای فیزیکی می‌تواند به وسیله اختصاص هر چهار گره از شبکه مجازی به یک سرور فیزیکی بهینه شود، که توسط آن پهنای باند خروجی مورد نیاز هر قفسه کمینه می‌شود. به این معنی که از دست رفتن پهنای باند بدون اختصاص پهنای باند بیش از حد در ناحیه هسته شبکه فیزیکی کمینه می‌شود. این راه‌حل برای یک مرکز داده به تنهایی ارائه شده است. هزینه منابع و کارایی می‌تواند در یک محیط تشکیل شده از چندین مرکز داده و شبکه‌های گسترده متفاوت باشد.

4- مقایسه

این بخش این پیشنهادها را با استفاده از مجموعه‌ای از معیارهای کیفی مقایسه می‌کند. هر پیشنهاد با استفاده از پنج معیار زیر ارزیابی شده است: قابلیت توسعه، تحمل خطا، قابلیت پیاده‌سازی، پشتیبانی از کیفیت سرویس و تعادل بار. قابلیت توسعه و تحمل خطا نگرانی‌های مهم طراحی برای مراکز داده دارای تعداد زیاد سرورها و منابع شبکه هستند و انتظار پشتیبانی از تعداد زیاد برنامه‌های کاربردی مستاجر می‌رود. چون مراکز داده عموماً از سرورها و سخت‌افزار امروزی شبکه استفاده می‌کنند، قابلیت پیاده‌سازی موضوع کلیدی است که نگران چگونگی تغییرات زیاد مورد نیاز در زیرساخت برای اجرای یک معماری خاص است. کیفیت سرویس دغدغه فزاینده مستاجران است و برای موفقیت در معماری مرکز داده مجازی مهم است. در نهایت، تعادل بار هدف مهمی از اپراتورهای شبکه برای مهندسی ترافیک و کمینه کردن ازدحام در شبکه‌های مرکز داده است. ما نتایج مقایسه خود را در جداول 2-6 خلاصه کرده‌ایم، هر جدول پیشنهاد ها را با استفاده از معیار خاصی از ویژگی مشخص مقایسه کرده است.

جدول 2: مقایسه کیفی طرحهای ارسال‌کننده

پیشنهاد	قابلیت توسعه	تحمل خطا	قابلیت استقرار	کیفیت سرویس	تعادل بار
مرکز داده سنتی	کم	خیر	زیاد	خیر	خیر
Diverter	زیاد	بله	زیاد	خیر	خیر
NetLord	زیاد	خیر	کم	خیر	بله
VICTOR	کم	بله	کم	خیر	خیر
VL2	زیاد	بله	کم	خیر	بله
PortLand	زیاد	بله	کم	خیر	بله
SecondNet	زیاد	بله	زیاد	بله	خیر
SEC2	کم	خیر	کم	خیر	خیر
CloudNaaS	کم	بله	کم	بله	خیر
Zeppelin	کم	بله	زیاد	خیر	بله

جدول 3: مقایسه کیفی پیشنهادها با توجه به چندمسیری

پیشنهاد	قابلیت توسعه	تحمل خطا	قابلیت استقرار	کیفیت سرویس	تعادل بار
مرکز داده سنتی	کم	خیر	زیاد	خیر	خیر
SPAIN	کم	بله	زیاد	خیر	بله
NetLord	زیاد	خیر	کم	خیر	بله
VL2	زیاد	بله	کم	خیر	بله
PortLand	زیاد	بله	کم	خیر	بله
تحمل خطا در معماری مجازی‌سازی شبکه مرکز داده	زیاد	بله	زیاد	بله	بله

جدول 4: مقایسه کیفی پیشنهادها با توجه به پهنای باند تضمین شده

پیشنهاد	قابلیت توسعه	تحمل خطا	قابلیت استقرار	کیفیت سرویس	تعادل بار
Oktopus	زیاد	بله	زیاد	بله	خیر
SecondNet	زیاد	بله	زیاد	بله	خیر
Gatekeeper	زیاد	بله	زیاد	بله	خیر
CloudNaaS	کم	بله	کم	بله	خیر

جدول 5: مقایسه کیفی پیشنهادها با توجه به اشتراک پهنای باند

پیشنهاد	قابلیت توسعه	تحمل خطا	قابلیت استقرار	کیفیت سرویس	تعادل بار
Seawall	زیاد	بله	زیاد	خیر	خیر
NetShare	کم	بله	کم	خیر	بله
اختصاص شبکه مجازی برای تحمل خطا با کارایی پهنای باند	کم	بله	کم	خیر	خیر

جدول 6: مقایسه کیفی پیشنهادها با توجه به تحمل خطا

پیشنهاد	قابلیت توسعه	تحمل خطا	قابلیت استقرار	کیفیت سرویس	تعادل بار
Zeppelin	کم	بله	زیاد	خیر	بله
تحمل خطا در معماری مجازی‌سازی شبکه مرکز داده	زیاد	بله	زیاد	بله	بله
اختصاص شبکه مجازی برای تحمل خطا با کارایی پهنای باند	کم	بله	کم	خیر	خیر

مقایسه ما از معماری‌های پیشنهادی مختلف چندین مشاهده را نشان می‌دهد. یک، هیچ راه‌حل ایده‌آلی برای تمام موضوعاتی که باید در زمینه مجازی‌سازی شبکه مرکز داده بررسی شود وجود ندارد. این موضوع عمدتاً به خاطر این است که هر معماری تلاش می‌کند تا روی جنبه خاصی از مجازی‌سازی مرکز داده تمرکز کند. از طرف دیگر، ترکیب ویژگی‌های کلیدی بعضی از معماری‌ها برای سود بردن از مزایای مربوط به آنها ممکن است. برای مثال، ترکیب VICTOR و Oktopus برای استقرار مرکز داده مجازی با تضمین پهنای باند ممکن است، درحالیکه پشتیبانی کارآمد برای مهاجرت ماشین مجازی ارائه می‌دهد. دوم، پیدا کردن بهترین معماری (یا ترکیب) نیاز به درک دقیق نیازهای کارایی از برنامه‌های کاربردی‌ای دارد که در مرکز داده قرار دارند. بنابراین، موضوعات بحث به تلاش‌های تحقیقاتی بیشتری در زمینه محیط‌های مختلف مرکز داده دارند.

5- نتیجه گیری

مراکز داده یک زیرساخت کارا از نظر هزینه برای ذخیره داده و میزبانی برنامه‌های کاربردی شبکه با مقیاس بزرگ هستند. در حالیکه، معماری‌های شبکه مرکز داده سنتی برای محیط‌های مرکز داده چند مستاجره آینده مناسب نیستند. مجازی‌سازی تکنولوژی امیدوار کننده ایی برای طراحی مراکز داده قابل توسعه و ساده در پیاده‌سازی است که انعطاف‌پذیری نیاز برنامه‌های کاربردی مستاجر را تامین می‌کند در حالیکه هزینه زیرساخت را کاهش داده، انعطاف‌پذیری مدیریت را بهبود می‌دهد و مصرف انرژی را کاهش می‌دهد.

در این مقاله، ما تحقیقات جدید و نو مجازی‌سازی شبکه مرکز داده را بررسی کرده‌ایم. طرح‌های پیشنهادی را از منظرهای مختلف مورد بحث قرار دادیم و روند های تحقیقاتی را در طراحی این معماری‌ها برجسته کردیم. گرچه پیشنهادهای فعلی قابلیت توسعه را بهبود می‌بخشد، مکانیزم‌هایی را برای تعادل بار و اطمینان از تضمین پهنای باند ارائه می‌دهد، چالش‌ها و موضوعات مهمی وجود دارد که هنوز باید مورد بررسی قرار گیرد. طراحی شبکه‌های هوشمند لبه، ارائه تضمین های محکم عملکرد، ایجاد مدل‌های قیمت گذاری و تجاری کارا، اطمینان از امنیت و قابلیت برنامه‌ریزی، پشتیبانی از زیرساخت های مرکز داده چند مکانی و چند لایه‌ای، اجرای تامین و مدیریت انعطاف‌پذیر رابط‌ها بین مستاجران و ارائه دهنده‌ها و استقرار ابزارهای کارا برای مدیریت مراکز داده مجازی‌شده جهات مهمی در تحقیقات آتی هستند.

مراجع

[1] A. Edwards, F. A, and A. Lain. (2009). Diverter: A New Approach to Networking Within Virtualized Infrastructures. ACM WREN.
[2] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta. (2009). VL2: A Scalable and Flexible Data Center Network. ACM SIGCOMM.
[3] A. Shieh, S. Kandulaz, A. Greenberg, C. Kim, and B. Saha. (2011). Sharing the Data Center Network. USENIX NSDI.

[4] C. Guo, G. Lu, H. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang. (2010). SecondNet: A Data Center Network Virtualization Architecture with Bandwidth Guarantees. ACM CoNEXT.
[5] C. Leiserson. (1985). Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing. IEEE Trans. Comput., vol. 34, no. 10, pp. 892–901.
[6] D. Carr. (2006). How Google Works.
[7] E. Rosen, A. Viswanathan, and R. Callon. (2001). Multiprotocol Label Switching Architecture. IETF RFC 3031.
[8] E. Rosen and Y. Rekhter. (2006). BGP/MPLS IP Virtual Private Networks (VPNs). IETF RFC 4364.
[9] F. Hao, T. Lakshman, S. Mukherjee, and H. Song. (2009). Enhancing Dynamic Cloud-based Services using Network Virtualization. ACM VISA.
[10] F. Hao, T. Lakshman, S. Mukherjee, and H. Song. (2010). Secure Cloud Computing with a Virtualized Network Infrastructure. USENIX HotCloud.
[11] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron. (2011). Towards Predictable Datacenter Networks. ACM SIGCOMM.
[12] H. Rodrigues, J. R. Santos, Y. Turner, P. Soares, and D. Guedes. (2011). Gatekeeper: Supporting Bandwidth Guarantees for Multi-tenant Datacenter Networks. WIOV.
[13] J. Dean and S. Ghemawat. (2004). MapReduce: Simplified Data Processing on Large Clusters. USENIX OSDI.
[14]James Kempf, Ying Zhang, Ramesh Mishra and Neda Beheshti. (20 l3). Zeppelin - A Third Generation Data Center Network Virtualization Technology based on SDN and MPLS. IEEE 978-1-4799-0568-3/13/$3l.00.
[15] J. Mudigonda, P. Yalagandula, B. Stiekes, and Y. Pouffary. (2011). NetLord: A Scalable Multi-Tenant Network Architecture for Virtualized Datacenters. ACM SIGCOMM.
[16] J. Mudigonda, P. Yalagandula, M. Al-Fares, and J. Mogul. (2010). SPAIN:COTS Data-Center Ethernet for Multipathing over Arbitrary Topologies. ACM USENIX NSDI.
[17] M. Al-Fares, A. Loukissas, and A. Vahdat. (2008). A Scalable, Commodity Data Center Network Architecture. ACM SIGCOMM.
[18] M. Chowdhury and R. Boutaba. (2010). A Survey of Network Virtualization. Computer Networks, vol. 54, no. 5, pp. 862–876.
[19] M. Shreedhar and G. Varghese. (1996). Efficient Fair Queuing Using Deficit Round-Robin. IEEE/ACM Trans. Netw., vol. 4, no. 3, pp. 375–385.
[20] R. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat. (2009). PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric. ACM SIGCOMM.
[21]Sagar C. Joshi and Krishna M. Sivalingam. (2013). On Fault Tolerance in Data Center Network Virtualization Architectures. IEEE ANTS 1569796067.
[22] San Jose. (2004). Data Center: Load Balancing Data Center Services SRND. Cisco Systems, Inc.
[23] T. Benson, A. Akella, A. Shaikh, and S. Sahu. (2011). CloudNaaS: A Cloud Networking Platform for Enterprise Applications. ACM SOCC.
[24] T. Lam, S. Radhakrishnan, A. Vahdat, and G. Varghese. (2010). NetShare: Virtualizing Data Center Networks across Services. Technical Report CS2010-0957.
[25] W. Dally and B. Towles. (2004). Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers Inc.
[26]Yukio Ogawa, Go Hasegawa and Masayuki Murata. (2014). Virtual Network Allocation for Fault Tolerance with Bandwidth Efficiency in a Multi-Tenant Data Center. IEEE 6th International Conference on Cloud Computing Technology and Science.
[27] http://aws.amazon.com/ec2/. Amazon Elastic Compute Cloud (Amazon EC2).