The Creation of an Infrastructure as a Service: Experiences and Challenges

[1] Amir Ziashahabi, [2]Mehrdad Ashtiani

[1] Iran University of science and technology, Tehran, Iran
amirziashahabi@comp.iust.ac.ir
[2] Assistant Professor, Iran University of science and technology, Tehran, Iran
m_ashtiani@iust.ac.ir

Abstract

Cloud computing provides its users the ability to leverage computational resources based on their demand and according to their requirements. One of the biggest advantages of such an approach is that the waste in computational resources would be reduced to a minimum. In order to provide a proper response to the needs of its laboratories, computer engineering department of Iran University of science and technology has decided to create an infrastructure as a service center well-known in the cloud computing paradigm. At its core, the center uses Openstack as one of the most widely-accepted and commonly used open source platforms for managing infrastructural resources. In this paper, we will review the strategic decisions being made by the major contributors of the center. To this end, we will first discuss about the fundamental concepts in cloud computing and move on to exchange views on the pipelines, challenges, and experiences gained throughout the process.

Keywords: Cloud computing, Infrastructure as a service, Openstack

Archive of SID

www.SID.ir

4th International Conference on Web Research
April 25-26, 2018; Tehran, IRAN
ICWR2018
کنفرانس بین المللی وب پژوهی
چهارمین
۵ و ۶ اردیبهشت ۱۳۹۷؛ تهران، ایران

ایجاد یک «زیرساخت به عنوان خدمت»: تجربه‌ها و چالش‌ها

امیر ضیاشهابی[۱]، مهرداد آشتیانی[۲]

[۱] دانشکده کامپیوتر، دانشگاه علم و صنعت ایران ، تهران
amirziashahabi@comp.iust.ac.ir

[۲] استادیار، دانشکده کامپیوتر، دانشگاه علم و صنعت ایران ، تهران
m_ashtiani@iust.ac.ir

Archive of SID

چکیده

مدل رایانش ابری برای کاربران خود این امکان را فراهم می‌آورد که به فراخور تقاضا، منابعی که دقیقاً متناسب با نیاز آنها است را در اختیار بگیرند. یکی از مهمترین مزایای این روش آن است که هدر رفتن منابع به حداقل خود می‌رسد. دانشکده کامپیوتر دانشگاه علم و صنعت برای پاسخگویی به نیاز آزمایشگاه‌های خود به منابع پردازشی اقدام به راه‌اندازی مرکزی بر پایه مدل رایانش ابری «زیرساخت به عنوان خدمت» نموده است. این مرکز، در زیرساخت خود از سیستم اپن استک که از معروف‌ترین و پرکاربردترین نرم افزارهای مدیریت زیرساخت به صورت منبع باز است استفاده می‌کند. در این مقاله، به بررسی رویکرد انتخاب شده توسط مرکز رایانش ابری دانشکده کامپیوتر دانشگاه علم و صنعت ایران برای ارایه خدمات خود می‌پردازیم. در این راستا، ابتدا مفاهیم رایانش ابری مطرح شده و سپس چالش ها و تجربیات کسب شده در مرکز رایانش ابری دانشکده کامپیوتر دانشگاه علم و صنعت مورد بررسی قرار خواهد گرفت.

کلمات کلیدی

رایانش ابری، زیرساخت به عنوان خدمت، اپن‌استک

۱- مقدمه

روند جاری در حوزه سامانه‌های کامپیوتری به سمتی پیش می‌رود که کاربران در عوض راه‌اندازی خدمات مورد نیاز خود بر روی مراکز داده سنتی، آنها را از ارایه‌دهندگانی تهیه نمایند که این خدمات را از طریق اینترنت عرضه‌کرده و وظیفه راه‌اندازی و نگهداری آنها را نیز بر عهده می‌گیرند. این مراکز دارای استخری از منابع هستند که توهم وجود منابع نامحدود را برای کاربران ایجاد می‌کند. همچنین، خدمات این ارایه‌دهندگان از قابلیت انعطاف بالایی بهره برده که در نتیجه آن، متقاضی می‌تواند دقیقاً منابع متناظر با نیاز خود را دریافت کرده و به همان میزان نیز هزینه پرداخت نماید. مفاهیم ذکر شده در واقع ویژگی‌های اصلی رایانش ابری را تشکیل می‌دهند. یکی از انواع خدمات رایانش ابری «زیرساخت به عنوان خدمت»[۱] است. در این نوع خدمت، یک ارایه‌دهنده خدمت، سخت‌افزار، سیستم عامل، مکان ذخیره‌سازی و سایر زیرساخت‌های لازم را برای کاربران فراهم می‌کند. اپن‌استک[۲]، یک بستر رایگان و منبع‌باز برای رایانش ابری است که برای راه‌اندازی «زیرساخت به عنوان خدمت» بکار می‌رود. مرکز رایانش ابری دانشکده کامپیوتر اقدام به استفاده از این بستر برای ارایه خدمات به آزمایشگاه‌های خود نموده است. در ادامه این مقاله مرکز یاد شده را با نام CCC[۳] خطاب می‌کنیم.

۱-۱- انگیزه‌های آغاز پروژه

از جمله مهترین مزایای استفاده از مدل ابری زیرساخت به عنوان خدمت می‌توان به طبیعت انعطاف‌پذیر این مدل برای اختصاص و آزادسازی میزان بسیار زیادی از منابع توسعه‌پذیر به محض درخواست اشاره نمود. همچنین، این مدل، امکان دسترسی به اطلاعات از هر مکانی را فراهم می‌کند که باعث می‌شود کاربران از محدودیت حضور فیزیکی در مکان منابع رهایی یابند.

دانشکده مهندسی کامپیوتر دانشگاه علم و صنعت ایران برای تجهیز آزمایشگاه‌های خود اقدام به تهیه منابع سخت‌افزاری نمود که بنا بود به صورت رایانه‌های جداگانه در اختیار آنها قرار گیرند. اما در ادامه، ایده راه‌اندازی یک «زیرساخت به عنوان خدمت» که تمامی آزمایشگاه‌ها بتوانند از آن استفاده نمایند مطرح شد. بدین صورت استخری از منابع ایجاد گردید که هر یک از آزمایشگاه‌ها می‌توانستند بسته به نیاز خود از آن بهره ببرند که در نتیجه آن، اتلاف منابع در آزمایشگاه‌ها نیز به حداقل خود می‌رسید.

www.SID.ir

۲- دانش پیش‌زمینه

در این بخش به ذکر مفاهیم اولیه مورد نیاز در زمینه رایانش ابری می‌پردازیم. سپس مولفه‌های موجود در اپن‌استک را مورد بررسی قرار خواهیم داد.

۲-۱- مفاهیم اولیه

در ابتدا لازم است که با مفهوم ابر و رایانش ابری در حوزه رایانه آشنا شویم. تعاریف مختلف و زیادی از رایانش ابری ارایه شده است. در این مقاله، تعریف پیشنهادی موسسه NIST[۴] را مورد بررسی قرار می‌دهیم.

طبق تعریفی که NIST ارایه می‌دهد، رایانش ابری مدلی برای برقراری دسترسی شبکه‌ای راحت، براساس تقاضا و قابل دسترسی از هر مکانی به استخر مشترکی از منابع کامپیوتری قابل پیکربندی است که قادر است با سرعت و با حداقل تلاش و یا ارتباط با ارایه‌دهنده خدمت، آزاد شده و دراختیار قرار گیرد [1]. کاربران خدمات ابری می‌توانند با استفاده از مرورگر، تلفن-همراه، کامپیوترهای شخصی و سایر ابزارهای کامپیوتری به این خدمات دسترسی پیدا کنند. این در حالی است که تمامی داده‌ها و برنامه‌ها بر روی سرویس‌دهنده‌ای در مکانی دور قرار گرفته که تمامی عملیات سنگین مد نظر را انجام می‌دهد [2]. طبق این تعریف، مدل ابری دارای ۵ خصیصه ضروری است [1]:

۱. بر اساس تقاضا و خودیاور[۵]: بر اساس این ویژگی، کاربر قادر خواهد بود به طور یک طرفه و بدون احتیاج به تعامل با هیچ‌یک از ارایه‌دهندگان خدمت به طور خودکار منابع کامپیوتری مانند زمان سرویس‌دهنده و حافظه را در اختیار بگیرد.
۲. دسترسی شبکه‌ای گسترده: این روش دسترسی به قابلیت‌های مورد نیاز را طریق شبکه و با استفاده از یک سری مکانیزم استاندارد فراهم می-کند. این مکانیزم‌ها بهبود استفاده از طریق بسترهای کاربری مختلف (مانند تلفن همراه، تبلت، لپ‌تاپ و غیره) را به همراه می‌آورند.
۳. ایجاد استخر منابع: منابع محاسباتی ارایه‌دهنده به صورت استخری برای خدمت‌رسانی به چندین مشتری در یک مدل چندمستأجری درمی‌آیند که در آن، منابع فیزیکی و مجازی به صورت پویا و بر اساس نیاز مشتری اختصاص داده شده و آزاد می‌شوند. در این مدل، گونه‌ای از استقلال از مکان وجود دارد، به این صورت که مشتریان عموماً کنترل یا اطلاعی از مکان دقیق منابع ارایه‌دهنده ندارند اما در سطح بالاتری از انتزاع(مانند شهر یا کشور) می‌توانند مکان را نیز تعیین کنند. به عنوان نمونه‌ای از این منابع می‌توان به فضای ذخیره‌سازی، حافظه، پردازنده و پهنای باند شبکه اشاره نمود.
۴. انعطاف سریع: قابلیت‌ها باید به صورت انعطاف‌پذیر قابل آزادسازی یا فراهم‌سازی (در بعضی موارد به صورت خودکار) باشد. در نتیجه، کاربر قادر خواهد بود براساس نیاز خود به سرعت منابع خود را توسعه یا کاهش دهد. از دیدگاه کاربر، منابع قابل فراهم‌سازی نامحدود هستند و در هر زمانی و به هر مقداری می‌توانند اختصاص داده شوند.
۵. خدمت اندازه‌گیری شده: سیستم ابری با استفاده از نوعی قابلیت اندازه-گیری که دارای سطح انتزاعی متناسب با نوع خدمت باشد به طور خودکار استفاده از منابع را کنترل و بهینه‌سازی می‌نماید. این قابلیت اندازه‌گیری معمولاً به صورت «براساس استفاده بپرداز» یا «براساس استفاده شارژ کن» است. بدین وسیله منابع می‌توانند کنترل و نظارت شوند که در نتیجه شفافیت هم برای فراهم کننده خدمت و هم برای کاربر فراهم می‌شود.

۲-۲- مدل‌های خدمت

خدمات ابری به طور کلی به سه دسته تقسیم می‌شوند [1]:

۱. «نرم‌افزار به عنوان خدمت» (SaaS): در این مدل، قابلیتی که در اختیار کاربر قرار می‌گیرد نرم‌افزار ارایه‌دهنده خدمت است که بر روی زیرساخت ابری اجرا می‌شود. برنامه از طریق ارتباط با یک رابط کاربری سبک وزن مانند مرورگر وب یا یک رابط نرم‌افزاری پیچیده قابل دسترسی است. کاربر توانایی تغییر هیچ یک از زیرساخت‌های ابری برنامه از جمله شبکه، حافظه، سیستم‌عامل و حتی بسیاری از ویژگی‌های برنامه را به جز ویژگی‌های محدودی که به عنوان «قابل تنظیم توسط کاربر» مشخص می‌شوند ندارد.
۲. «بستر به عنوان خدمت» (PaaS[۶]): در این مدل، قابلیتی که در اختیار کاربر قرار می‌گیرد استقرار نرم‌افزاری است که کاربر با استفاده از زبان-های برنامه‌نویسی، کتابخانه‌ها و خدمات و ابزارهایی که ارایه‌دهنده خدمت از آنها پشتیبانی می‌کند میتواند بستر خود را ایجاد نماید. کاربر کنترلی بر روی زیرساخت بکار رفته مانند شبکه، حافظه، سرویس دهنده‌ها و سیستم‌عامل ندارد. اما قادر است بر روی برنامه استقرار یافته و محیط میزبانی آن تنظیماتی را اعمال نماید.
۳. «زیرساخت به عنوان خدمت» (IaaS): قابلیت ارایه شده در این مدل، شامل توان پردازشی، حافظه، شبکه و تمام منابع کامپوتری پایه است که کاربر می‌تواند با استفاده از آنها برنامه مورد نظر خود شامل برنامه کاربردی و سیستم‌عامل دلخواه خود را مستقر و اجرا نماید. با این وجود، کاربر هیچ‌گونه کنترلی بر روی زیرساخت ابری بکار رفته ندارد. اما بر روی سیستم عامل، حافظه، برنامه استقرار یافته و احتمالاً به طور محدود مؤلفه‌های شبکه، کنترل دارد. خدمت، مالکیت زیرساخت را بر عهده داشته و مسئول میزبانی، راه‌اندازی و نگهداری آن است. پرداخت هزینه IaaS می‌تواند بر اساس یک قرارداد یا به صورت «به میزان استفاده بپرداز» باشد [3].

مرکز خدمات ابری CCC خدمات خود را به صورت «زیرساخت به عنوان خدمت» در اختیار کاربران قرار می‌دهد.

۲-۳- مروری بر بسترهای ابر IaaS موجود

۲-۳-۱- اکالیپتوس[۷]

اکالیپتوس یک بستر رایانش ابری ترکیبی است که توسط اکالیپتوس سیستمز توسعه داده شده است. این بستر در سال ۲۰۰۸ پایه‌گذاری و در سال ۲۰۱۴ به مالکیت HP درآمد. اکالیپتوس به عنوان یک راه‌حل IaaS ترکیبی که با وب سرویس‌های آمازون سازگاری بستری[۸] دارد، توسعه داده شده است. این ویژگی باعث می‌شود که سازمان‌ها این امکان را داشته باشند که ابزارها، تصاویر[۹] و اسکریپت‌های سازگار با Amazon-AWS را برای مدیریت محیط اکالیپتوس استفاده نمایند [5]. اکالیپتوس به گونه‌ای طراحی شده است که نصب آن کاربرپسند[۱۰] باشد و احتیاجی به منابع اختصاصی یا شخصی-

Archive of SID
www.SID.ir

سازی‌شده نداشته باشد. این بستر منبع‌باز است و چارچوب آن به خوبی پیمانه-بندی شده است [6].

۲-۳-۲- اپن‌نبولا[11]

اپن‌نبولا یک بستر IaaS منبع باز است که در سال ۲۰۰۸ منتشر شد. هدف این ابزار آوردن زیبایی، آرامش ذهن و سادگی به ابرهای تجاری خصوصی و ترکیبی است [5]. اپن‌نبولا یک بستر متمرکز است و پروژه آن توسط یک سازمان رهبری می‌شود. این بستر شامل سه لایه می‌شود: ابزارها، گرداننده‌ها و هسته [5]. همچنین این بستر از قابلیت شخصی‌سازی بسیار بالایی برخوردار است [6]. بستر اپن‌نبولا از رابط‌های دسترسی مختلفی پشتیبانی می‌کند، از جمله رابط سرویس OCCI، رابط مبتنی بر REST و استاندارد-های نوظهور مربوط به API[12] در فضای ابری [7].

۲-۳-۳- آپاچی کلوداستک[13]

کلوداستک یک بستر منبع باز است که توسط Apache Software Foundation ایجاد شده است و برای ساختن، مدیریت و استقرار ابرهای IaaS مورد استفاده قرار می‌گیرد [6]. کلوداستک برای استقرار و مدیریت شبکه بزرگی از ماشین‌های مجازی و فراهم‌سازی ابرهای IaaS با قابلیت گسترش بالا طراحی شده است [6]. این بستر توسط بسیاری از ارایه‌دهندگان خدمات ابری برای ارایه خدمات ابری عمومی و توسط بسیاری دیگر برای ارایه خدمات ابری خصوصی با دسترسی از راه دور و یا به عنوان بخشی از یک راه‌حل ابر ترکیبی مورد استفاده قرار گرفته است [5].

۲-۳-۴- اپن‌استک

اپن‌استک یک بستر نرم‌افزاری برای ایجاد و مدیریت ابرهای خصوصی و عمومی است. این نرم‌افزار مجموعه‌ای از پروژه‌های متن باز است که شامل راه‌حلی برای ایجاد «زیرساخت به عنوان خدمت» با استفاده از مجموعه‌ای از خدمات وابسته به هم می‌شود. این پروژه توسط ناسا و رکاسپیس آغاز شده است [2,4]. بستر اپن‌استک بسیار گسترش‌پذیر است و می‌توان از آن برای توسعه معماری‌های توزیع شده با حداکثر ۱ میلیون ماشین فیزیکی و ۶۰ میلیون ماشین مجازی استفاده نمود [10]. شرکت‌های بسیاری در سراسر دنیا در توسعه این بستر شرکت دارند [6]. اپن‌استک سیستم پیچیده‌ای است و نیاز به دانش فنی بالایی دارد که این موضوع استفاده از آن را در شرکت‌های غیرفنی مشکل می‌کند [8].

۲-۳-۵- مؤلفه‌ها

اپن‌استک از مؤلفه‌های زیادی تشکیل شده است که هر یک وظیفه‌ای در ابر IaaS بر عهده دارند. به علاوه، اغلب با ارائه شدن هر نسخه جدید، مؤلفه‌های جدیدی نیز ارائه شده‌اند که یا در جهت بهبود کارایی پیشین بوده‌اند و یا یک قابلیت جدید عرضه نموده‌اند. بنابراین، لازم است به هنگام استفاده از اپن‌استک تنها آن مؤلفه‌هایی را در وهله‌ی اول نصب نمود که ما را به حداقل محصول قابل دوام و یا در اصطلاح MVP[14] برساند.

اکنون به اختصار به تشریح هر یک از این بخش‌ها می‌پردازیم [2]:

۱. محاسبه[15] (نوا[16]): مؤلفه نوا، کنترل‌کننده اصلی پردازش ابری در اپن-استک است. این مؤلفه مسئولیت استقرار و مدیریت تعداد بسیار زیادی از ماشین‌های مجازی و سایر نمونه[17]ها را برای انجام پردازش‌های کامپیوتری بر عهده دارد.

۲. خدمت شناسایی (کی‌استون[18]): کی‌استون خدمت شناسایی را برای اپن-استک فراهم می‌کند یا به عبارت دیگر این مؤلفه نقش یک کتاب راهنما را ایفا می‌کند که کاربران را به خدماتی از اپن‌استک که به آنها دسترسی دارند نگاشت می‌کند. این خدمت، به عنوان راه مشترک برای احراز هویت در تمام ابر بکار می‌رود که از طریق چندین روش می‌توان به آن دسترسی پیدا کرد.

۳. انبار اشیاء (سوییفت[19]): انبار اشیا یک حافظه افزونه توسعه‌پذیر برای اشیا و فایل‌ها است. اشیا و همچنین فایل‌ها در چندین دیسک در چندین سرویس‌دهنده از سرویس‌دهنده‌های موجود در مرکز داده نوشته می‌شوند. اپن‌استک، صرفاً وظیفه بررسی صحت و رونویسی داده در ابر را بر عهده دارد.

۴. انبار بلوک[20] (سیندر[21]): سیندر یک انبار بلوک است که کاربرد آن مشابه زمانی است که یک رایانه سنتی فایل مشخصی را بر روی دیسک سخت مورد دسترسی قرار می‌دهد. این مؤلفه، حافظه‌های پایداری را در سطح بلوک در اختیار نمونه‌های محاسباتی اپن‌استک قرار می‌دهد. در اپن‌استک، سیندر وظیفه مدیریت ساختن، پیوست کردن و جداکردن حافظه‌های بلوکی را در اختیار دارد.

۵. شبکه (نوترون[22]): این مؤلفه، قابلیت‌های شبکه را در اختیار اپن‌استک قرار می‌دهد و همچنین سیستمی برای مدیریت آسان، سریع و بهینه آدرس‌های آی‌پی است.

۶. داشبورد (هورایزن[23]): هورایزن داشبوردی است که رابط کاربری گرافیکی‌ای را در اختیار ادمین‌ها و کاربران اپن‌استک قرار می‌دهند که از طریق آن می‌توان دسترسی، فراهم‌سازی و همچنین خودکارسازی منابع ابری را انجام داد.

۷. خدمت تصویر (گلنس[24]): این مؤلفه، خدمات مربوط به تصویر را در اختیار خدمات اکتشاف، ثبت و تحویل که نیازمند تصاویر مربوط به دیسک و سرویس‌دهنده هستند، قرار می‌دهد. همچنین، این مؤلفه این امکان را فراهم می‌کند که این تصاویر برای استقرار نمونه‌های جدید از ماشین‌های مجازی، مورد استفاده قرار بگیرند.

۸. دورسنجی (سیلومیتر[25]): این مؤلفه این امکان را به اپن‌استک می‌دهند که برای اشخاصی که از خدمات آن بهره می‌برند صورتحساب تولید نماید. سیلومیتر میزان استفاده کاربر از هر یک از خدماتی که ارائه می-شوند را نگهداری می‌کند. این مؤلفه، داده‌های اندازه‌گیری شده بسیار زیادی از قسمت‌های مختلف اپن‌استک جمع‌آوری و نگهداری می‌کند. بحث اصلی در این پژوهش در ارتباط با این مؤلفه و اندازه‌ها است.

۹. هماهنگی (هیت[26]): هیت خدمتی است که به توسعه‌دهندگان این امکان را می‌دهد که نیازمندی‌های یک کاربرد ابری را در داخل یک فایل ذخیره نمایند که این فایل نشان می‌دهد که چه منابعی برای چه کاربردی مورد نیاز هستند. این مؤلفه، خدمتی را برای مدیریت چرخه زندگی زیرساخت و برنامه‌هایی که در داخل ابر بکار رفته‌اند فراهم می-کند. به صورتی که هم توسط انسان و هم ماشین قابل دسترسی باشد.

Archive of SID
www.SID.ir

هیت به همراهی سیلومیتر می‌توانند ویژگی خودتوسعه را برای ابر فراهم بیاورند.

۲-۳-٦- انواع گره در اپن استک

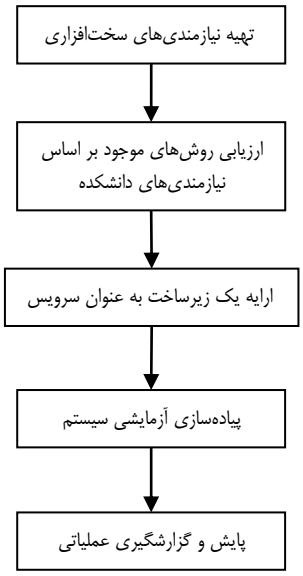
بستر اپن‌استک دارای دونوع گره است: کنترلر و محاسبه.

۱. گره محاسبه: گره محاسبه گره‌ای است که یک هایپروایزر[۳۷] و خدمت nova-compute را اجرا می‌کند. این گره همان جایی است که نمونه-های ماشین مجازی اجرا می‌شوند.
۲. گره کنترلر: گره کنترل در واقع شامل هر چیز دیگر می‌شود. از جمله رابط وب، زمان‌بند، سرویس‌دهنده پایگاه‌داده و صف پیام.

با وجود استفاده از این عبارات برای بیان گره‌های اپن‌استک لزوماً استقرار مؤلفه‌های بستر به این شکل نیست و می‌توان این مؤلفه‌ها را بر روی گره‌های مختلف توزیع نمود.

۲-٤- مرکز رایانش ابری دانشکده کامپیوتر دانشگاه علم و صنعت ایران

در این بخش به بررسی رویکردی می‌پردازیم که CCC برای راه‌اندازی یک «زیرساخت به عنوان خدمت» بکار برده است. شکل (۱) فرایند طی-شده جهت راه‌اندازی و بهره‌برداری مرکز یاد شده را نشان می‌دهد. در شکل (۲) نمایی از زیرساخت سخت افزاری «زیرساخت به عنوان سرویس» در دانشکده کامپیوتر دانشگاه علم و صنعت نشان داده شده است.



شکل (۱) : فرایند راه‌اندازی و بهره‌برداری CCC

۲-٤-۱- بستر اپن‌استک

در بخش ۲-۳ به معرفی اپن‌استک پرداختیم. در این بخش، به دلایل استفاده CCC از این بستر برای ارایه خدمات خود (با وجود بستر های دیگر از جمله اکالیپتوس، اپن نبولا و آپاچی کلوداستک) می‌پردازیم.

شکل(۲): نمایی از زیرساخت سخت‌افزاری ایجاد شده

www.SID.ir

۱. IBM، RackSpace، Dell، Cisco، Yahoo، VMWare، Intel و HP اشاره نمود. نام بیش از ۲۰۰ در شرکت در بین حامیان و پشتیبان‌های اپن‌استک مشاهده می‌شود [8, 14].
۲. مقرون به صرفه بودن: اپن‌استک امکانات یک بستر تجاری مشابه را به کسری از هزینه تهیه آن ارایه می‌دهد.
۳. قابلیت گسترش بالا: اپن‌استک قابلیت پشتیبانی از میلیون‌ها ماشین فیزیکی و مجازی را داراست [10].
۴. پشتیبانی از هایپروایزرهای مختلف: اپن‌استک توانایی پشتیبانی از محدوده وسیعی از هایپروایزرها را داراست از جمله: ESX، KVM، Xen، XenServer، HyperV، QEMU و LXC [11].
۵. کارایی: از دیگر برتری‌های اپن‌استک نسبت به سایر بسترها به خصوص اپن‌نبولا می‌توان به کارایی بالاتر آن اشاره نمود [5].

۲-٤-۲- چالش‌ها

در این بخش به ذکر مجموعه‌ای از مشکلات و چالش‌هایی می‌پردازیم که CCC در راستای ارایه خدمات بر روی بستر ابری با آنها مواجه شده است.

۱. محدودیت آدرس‌های آی‌پی

یکی از اصلی‌ترین موانع در راه ارایه خدمات با کیفیت توسط CCC مربوط به محدودیت آدرس‌های آی‌پی است. مجموعه آدرس‌های IP اختصاصی شامل ۶۴ آدرس می‌شد که تعدادی از آنها برای مواردی مانند درگاه‌های مسیریاب هر پروژه و آدرس آی‌پی گره کنترلر برای ایجاد امکان برقراری ارتباط از راه دور، مصرف شدند. بنابراین، تعداد کمی آدرس برای اختصاص به ماشین‌های مجازی باقی ماند. این موضوع به این معنی است که برای تعداد بسیار کمی ماشین، امکان برقراری ارتباط مستقیم از راه دور وجود دارد. این شرایط ایجاب می‌کند که برای هر یک از پروژه‌های موجود در اپن‌استک در استفاده از آدرس‌های آی‌پی سهمیه‌ای تعیین شود. این کار با استفاده از قابلیت آی‌پی شناور[۲۸] در اپن‌استک میسر می‌شود. با استفاده از این ویژگی استخری از آدرس‌های آی‌پی ایجاد می‌شود. سپس به هر یک از پروژه‌ها سهمی از این آدرس‌ها اختصاص داده می‌شود.

برای ایجاد امکان استفاده از این قابلیت لازم است که در هنگام ایجاد شبکه در اپن‌استک برای ساختن شبکه اصلی متصل به اینترنت در قسمت Provider Network، مقدار Network Type، flat و مقدار Physical Network، provider تعیین شود. هر کاربر در صورتی که نیاز به دسترسی مستقیم به یک ماشین مجازی داشته باشد لازم است که وارد محیط هورایزن شده و یک آدرس آی‌پی را به ماشین مورد نظر پیوست کند. پس از انجام اینکار، امکان برقراری ارتباط مستقیم با ماشین فراهم می‌شود. پس از برقراری ارتباط با یک ماشین می‌توان از طریق آن اقدام به برقراری ارتباط با سایر ماشین‌های مجازی داخل شبکه نمود. لازم به ذکر است که تمامی ماشین‌ها در پشت یک شبکه NAT شده قرار دارند. بنابراین، امکان برقراری ارتباط با اینترنت برای تمامی آنها برقرار است اما نمی‌توان از خارج به آنها متصل شد.

۲. عدم آشنایی افراد با شیوه استفاده از خدمات

مشکل عمده بعدی آشنا نبودن کاربران با شیوه استفاده از پنل هورایزن و ماشین‌های مجازی لینوکسی است. همچنین باید در نظر داشت که ارایه خدمات CCC محدود به دانشکده کامپیوتر نمی‌شود و ممکن است افرادی از دانشکده‌های دیگر مانند برق نیز برای اجرای پردازش‌های خود از خدمات این مرکز استفاده نمایند. این مسایل نیاز به آموزش شیوه استفاده از خدمات را به یک نیاز اساسی تبدیل می‌کند.

۳. قطعی برق

یکی از مواردی که مخصوصاً در تابستان باعث ایجاد مشکلات بسیاری شد قطعی ناگهانی برق بود که باعث می‌گردید تمامی گره‌ها در وضعیت نامطمئن خاموش شوند. این مسئله، علاوه بر ضربه زدن به فرایندهای در حال اجرای کاربران و ماشین‌های مجازی آنها، لزوم برقراری تماس مستقیم با ماشین‌های فیزیکی برای راه‌اندازی مجدد بستر را باعث می‌شد که بسته به شرایط، ممکن بود عملی زمانبر باشد. به علاوه، بستر در بسیاری از موارد پس از راه‌اندازی مجدد وارد حالتی می‌شد که در آن، بعضی از خدمات اپن‌استک غیر قابل دسترسی بودند. در نهایت، تهیه یک منبع تغذیه بدون وقفه یا به اختصار UPS در دستور کار قرار گرفت. در این رابطه، ایده‌آل آن است که UPS به تمامی گره‌ها متصل شده و در صورت قطع شدن برق تا زمانی که برق دوباره وصل شود گره‌ها را در حالت عادی نگه دارد. با توجه به هزینه بسیار بالا برای تهیه چنین UPS ای، تصمیم بر آن شد که یک منبع برای روشن نگه داشتن گره‌ها به مدت محدود تهیه شود. پس از قطعی برق، دستگاه UPS با استفاده از پروتکل SNMP به گره کنترلر اطلاع می‌دهد. کنترلر در این مدت محدود می‌تواند اقدام به خاموش کردن هر یک از گره‌های دیگر با حفظ حالت آنها نموده و با استفاده از پیامک یا ایمیل، مسئول نگه‌داری زیرساخت را از قطعی برق مطلع سازد.

۲-٤-۳- خط لوله‌های CCC

در این بخش به توضیح خط لوله‌هایی که CCC برای ارایه خدمات خود به کاربران استفاده می‌کند، می‌پردازیم.

۱. خط لوله اضافه کردن گره محاسبه

گره محاسبه، ماشینی فیزیکی است که ماشین‌های مجازی در واقع بر روی آن اجرا می‌شوند. در نتیجه، زمانی که یک ماشین مجازی اجرا می‌شود، منابع مورد نیاز آن توسط گره‌های محاسباتی فراهم خواهد شد. برای اضافه کردن یک گره محاسبه دو راه وجود دارد. راه نخست، نصب هر یک از مؤلفه‌های اپن‌استک که قصد استفاده از آنها را داریم بوده و راه حل دوم همتاسازی یکی از گره‌های محاسبه تنظیم شده بر روی یکی دیگر از گره‌های محاسبه است. در ادامه، هر یک از این رویکردها را مورد بررسی قرار می‌دهیم.

أ. نصب مؤلفه‌های اپن‌استک

نخستین راهی که برای اضافه نمودن یک گره محاسبه به ذهن می‌رسد پیمودن مراحل ذکر شده در مستندات اپن‌استک و نصب و تنظیم قدم به قدم مؤلفه‌ها است. برای اینکار لازم است ابتدا مؤلفه نوا که مسئولیت فراهم‌سازی ماشین‌های مجازی را دارد و مؤلفه‌ی نوترون که مسئولیت امور مربوط به شبکه را عهده‌دار است نصب شوند. سپس، بسته به ویژگی‌هایی که ابر دارا است مؤلفه‌های مورد نیاز را انتخاب و اقدام به نصب آن‌ها نماییم. برای مثال، در CCC بر روی هر یک از گره‌ها علاوه بر مؤلفه نوا دو مولفه دیگر نیز نصب شدند: (۱) مؤلفه‌ی سیندر برای ایجاد امکان استفاده از دیسک سخت موجود در گره و (۲) سیلومیتر برای فراهم کردن خدمت پایش بر روی گره. اما، مشکل اصلی که این روش دارد حجم زیادی از کار تکراری است که لازم است بر روی هر یک از گره‌ها انجام شود. برای رفع این مشکل راه‌حل همتا-سازی مطرح می‌شود که در ادامه به آن می‌پردازیم.

Archive of SID
www.SID.ir

ب. همتاسازی

روش دوم که توسط CCC نیز مورد استفاده قرار می‌گیرد استفاده از نرم-افزارهای همتاساز برای کپی کردن یک گره بر روی گرهی دیگر است. یکی از این نرم‌افزارها، کلون‌زیلا است.

کلون‌زیلا یک نرم‌افزار تصویرگیری و همتاسازی از روی دیسک است. این نرم‌افزار امکان پشتیبانی‌گیری و بازیابی در سطح سخت‌افزار خالص را در اختیار می‌گذارد. کلون‌زیلا دارای دو نوع است: (۱)زنده و (۲) نسخه سرویس دهنده. نسخه سرویس‌دهنده برای مستقرسازی در مقیاس بزرگ (تا ۴۰ رایانه به طور همزمان) مورد استفاده قرار می‌گیرد. نسخه زنده توسط CCC مورد استفاده قرار گرفت. این نسخه یک توزیع کوچک از گنو/لینوکس است. این نرم‌افزار باید بر روی ماشین مورد نظر در حالت Boot اجرا شود. سپس، با انجام دستوراتی در ماشین مقصد و ماشین مبدا عمل همتاسازی شروع می‌شود.

۲. خط لوله بروزرسانی اپن‌استک

دوگونه بروزرسانی برای اپن‌استک قابل انجام است. گونه اول که بروزرسانی کوچک نیز به آن گفته می‌شود به راحتی و در سیستم‌عاملی مانند اوبونتو با دستور sudo apt-get upgrade PACKAGENAME قابل انجام است. معمولاً این نوع بروزرسانی‌ها برای رفع باگ‌های مربوط به نسخه موجود انجام می‌شوند. اما گونه‌ای دیگر از بروزرسانی نیز وجود دارد که برای ارتقاء اپن‌استک از نسخه‌ای به نسخه دیگر انجام می‌شود. این گونه بروزرسانی، بروزرسانی بزرگ نامیده می‌شود. انجام این نوع بروزرسانی دارای چالش‌های زیادی است. برای انجام این بروزرسانی ابتدا لازم است که مخزن دریافت بسته‌های اپن‌استک به مخزن مربوط به نسخه جدید بروزرسانی شود. سپس هر یک از مؤلفه‌های موجود از مخزن جدید دوباره نصب شوند که معمولاً این عمل به دلیل عدم سازگاری مؤلفه‌های نسخه جدید با نسخه پیشین موجب غیر قابل استفاده شدن زیرساخت می‌شود. پس لازم است که به مستندات هر یک از مؤلفه‌ها رجوع گردیده و تک تک قسمت‌هایی که در فایل‌های کانفیک تغییر کرده‌اند شناسایی شده و تغییرات بر روی فایل کانفیک اعمال شوند. این فرایند عملی دشوار و بسیار زمانبر است. همچنین، معمولاً در هر نسخه اپن‌استک شمای پایگاه‌داده تغییر می‌کند. تغییر شمای کنونی و سازگار نمودن آن با شمای جدید عملی مشکل و در بسیاری موارد غیر ممکن است. در نتیجه برای اعمال شمای جدید ممکن است مجبور به حذف پایگاه‌داده کنونی و ایجاد پایگاه‌داده جدیدی شویم که در نتیجه، تمامی اطلاعات موجود نیز به همراه پایگاه‌داده حذف می‌شوند. بنا به دلایل ذکر شده انجام بروزرسانی‌های بزرگ مثل گذر از نسخه نیوتون به اکاتا کار بسیار مشکلی است و پیشنهاد نمی‌شود.

۳. خط لوله اضافه کردن گره گرافیکی

بسیاری از کاربران برای انجام پردازش‌های خود نیاز به ماشینی دارند که از کارت گرافیک قدرتمند بهره ببرد. مرکز CCC در زمان نگارش این پژوهش دارای چندین ماشین مجهز به کارت گرافیک قدرتمند است. یک ماشین مجهز به یک کارت گرافیک Nvidia 1080، ماشین دیگری مجهز به دو کارت گرافیک Nvidia 1080 که به صورت SLI به یکدیگر متصل شده‌اند و همچنین سه ماشین هر کدام مجهز به دو کارت گرافیک Nvidia 950 که آنها نیز به صورت SLI به یکدیگر متصل شده‌اند. اما چالش اصلی استفاده از چنین ماشین‌هایی آن است که بستر اپن‌استک و هایپروایزر KVM به طور پیشفرض حالتی را برای ارایه چنین ماشین‌هایی در نظر نگرفته‌اند. بدین صورت که هایپروایزر به طور پیشفرض قادر به برقراری ارتباط مابین ماشین مجازی و کارت گرافیک کار گذاشته شده در ماشین فیزیکی نیست. برای رسیدن به این امکان راه‌حل‌های متعددی بررسی شد که هیچ‌یک از آنها نتیجه مطلوب را دربر نداشتند. بنابراین، تصمیم بر آن شد که بر روی ماشین-های دارای کارت گرافیک سیستم‌عامل لینوکس و همچنین گرداننده‌های مورد نیاز کارت گرافیک نصب شوند و تنظیمات آنها نیز انجام گیرد. سپس، امکان ارتباط متقاضیان با این ماشین‌ها از طریق ssh فراهم شود. شایان ذکر است که این روش دارای مشکلات امنیتی بزرگی است به طوری که کاربران می‌توانند با تغییر رمز عبور عملاً امکان دسترسی به ماشین را از بین ببرد. برای حل این مشکل ابزار Free IPA مورد استفاده قرار گرفت. این ابزار امکان ایجاد کاربر و نگاشت آنها به ماشین‌های مجازی را فراهم می‌نمود. همچنین با استفاده از این ابزار می‌توان تعیین کرد که هر کاربر امکان اجرای چه دستوراتی را دارد. مثلا استفاده از دستور sudo را می‌توان محدود به نصب بسته‌های نرم‌افزاری نمود که این عمل امکان انجام دستوراتی مانند تغییر رمز عبور را از کاربران سلب می‌کند.

٤. درخواست ماشین مجازی

رویکرد CCC به این صورت است که منابعی که در زیرساخت وجود دارد تقسیم شده و به هر یک از آزمایشگاه‌ها سهمی اختصاص یابد. سپس، برای هر یک از آزمایشگاه‌ها مسئولی تعیین شود که مسئولیت مدیریت منابع و ماشین‌های مجازی آزمایشگاه را بر عهده گیرد. همچنین، به هر یک از آزمایشگاه‌ها یک پنل هورایزن نیز اختصاص می‌یابد که انجام امور مدیریتی مسئول از طریق این پنل و به طور مستقل از مسئولان زیرساخت را امکان پذیر سازد. درخواست‌های دریافت ماشین مجازی می‌توانند به دو دسته تقسیم شوند:

۱. دسته اول درخواست‌هایی هستند که مربوط به یکی از آزمایشگاه‌های دانشکده مهندسی کامپیوتر هستند. این درخواست‌ها توسط مسئول آزمایشگاه پاسخ داده می‌شوند.

۲. دسته دوم درخواست‌هایی هستند که از خارج از دانشکده مهندسی کامپیوتر داده می‌شوند. برای پاسخ به این درخواست‌ها CCC سامانه‌ای را تدارک دیده است. هر دانشجویی که نیاز به دریافت یک ماشین مجازی داشته باشد می‌تواند درخواست خود را در این سامانه ثبت کند. سپس، مسئول سامانه با توجه به درخواست دانشجو و منابع موجود اقدام به ایجاد ماشین مجازی می‌نماید و در نهایت یک آدرس آی‌پی در اختیار متقاضی قرار می‌گیرد که توسط ssh قابلیت اتصال به ماشین مجازی را به متقاضی خواهد داد. همچنین، لازم به ذکر است که برای اتصال به ماشین‌های مجازی از طریق ssh نیاز به یک کلید خصوصی است که با کلید عمومی که در اپن‌استک به ماشین مجازی مورد نظر پیوست شده است جفت باشد. این کلید توسط اپن‌استک ساخته شده و در اختیار کاربر قرار می‌گیرد. سپس، این کلید در مورد آزمایشگاه‌های داخل دانشکده مهندسی کامپیوتر، توسط مسئول آزمایشگاه و در مورد دانشجویان خارج از آزمایشگاه‌های ذکر شده توسط سامانه در اختیار متقاضی قرار می‌گیرد. این کلید می‌بایست به عنوان پارامتر به فرمان ssh اضافه شود. وجود

Archive of SID
www.SID.ir

این کلید برای ورود به ماشین مجازی الزامی است و امکان تولید مجدد آن وجود ندارد. مسئولیت نگهداری از کلید به عهده متقاضی است.

۵. خط لوله کاری برای پایش معیارهای کلیدی سیستم

در ابتدا لازم است که اشاره شود که پایش زیرساخت CCC دارای دو بخش است. بخش اول مربوط به پایش زیرساخت شامل گره کنترلر و گره‌های محاسبه و بخش دوم شامل پایش عملکرد و مصرف منابع توسط کاربران و ماشین‌های مجازی است.

برای پاسخ به بخش اول، سامانه زبیکس[۲۹] که یک ابزار منبع‌باز برای پایش زیرساخت است، استفاده شده است. از طرف دیگر مولفه‌ی سیلومیتر اپن‌استک وظیفه پایش کاربران را انجام می‌دهد. اما این مولفه فاقد رابط کاربری گرافیکی برای ارتباط با کاربران نهایی است. برای حل این مشکل، یک رابط گرافیکی تحت وب طراحی گردید. این رابط از API مولفه‌های کی استون، نوا و سیلومیتر در زبان برنامه‌نویسی پایتون برای بدست آوردن اطلاعات مربوط به هر کاربر استفاده می‌کند و سپس چارچوب جنگو[۳۰] را برای نمایش اطلاعات بکار می‌برد.

۳- نتیجه‌گیری

در این مقاله رویکرد مرکز رایاتش ابری دانشکده کامپیوتر دانشگاه علم و صنعت برای راه‌اندازی یک بستر بر پایه مدل «زیرساخت به عنوان خدمت» مورد بررسی قرار گرفت. در این مقاله در ابتدا به بیان مفاهیم پایه در رابطه با رایانش ابری و مدل‌های ارایه خدمت پرداختیم. سپس، اپن‌استک به عنوان بستری که برای ارایه خدمات در نظر گرفته شده است معرفی شد. در نهایت، رویکرد مرکز رایانش ابری دانشکده کامپیوتر با مرور دلایل انتخاب اپن‌استک، چالش‌های مرکز و خط لوله‌های آن مورد بررسی قرار گرفت.

مراجع

[1] P. Mell, T. Grace, The NIST Definition of Cloud Computing, *National Institute of Standards and Technology*, Special Publication,2011
[2] R. Kumar, N. Gupta, S. Charu, K. Jain and S. K. Jangir, Open Source Solution for Cloud Computing Platform using Openstack, *International Journal of Computer Science and Mobile Computing*, Vol.3, No.5, p. 90, 2014.
[3] S. Bhardwaj, L. Jain and S. Jain, Cloud computing: A study of Infrastructure as a Service (IaaS), *International Journal of Engineering and Information Technology*, Vol.2, No.1, pp. 61-63, 2010.
[4] OpenStack, *OpenStack Documentation*, August 2017, http://docs.openstack.org.
[5] M. Iyowu, Cloud Computing Comparative Study of OpenStack vs OpenNebula, *University of Portsmouth*, MS thesis, 2016.
[6] D. Freet, R. Agrawal, J. J. Walker and Y. Badr, Open Source Cloud Management Platforms and Hypervisor Technologies: A Review and Comparison, in *Proceedings of the Southeast Conferene*, Norfolk, VA, 2016, pp. 1-8
[7] S. Shahzadi, M. Iqbal, Z. Qayyum and T. Dagiuklas, Infrastructure as a Service (IaaS): A Comparative Performance Analysis of Open-source Cloud Platforms, in *Proceedings of the 2017 IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*,Barcelona, Spain, 17-19 September, 2017.
[8] R. Kumar, K. Jain, H. Maharwal, N. Jain and A. Dadhich, Apache CloudStack: Open Source Infrastructure as a Service Cloud Computing Platform, *International Journal of Advancement in Engineering Technology, Management & Applied Science*, pp. 111-116, 2014.
[9] N. Serrano, G. Gallardo and J. Hernantes, Infrastructure as a Service and Cloud Technologies, *IEEE Software*, vol. 32, no. 2, pp. 30-36, 2015.
[10] K. Pepple, Deploying OpenStack, *O'Reilly Media, O'Reilly Publishing*, 2015.
[11] P.Takako, G.Esticio, J.Kelner and D.Sadok, A Survey on Open-source Cloud Computing Solutions, in *Proceedings of the 8th Workshop on Clouds, Grids and Applications*, pp. 142-150, 2010.
[12] Q.Jiang, Open Source IaaS Community Analysis, OpenStack vs OpenNebula vs Eucalyptus vs CloudStack, Jilin University, School of Information Technology, Msc Thesis, 2014
[13] X. Wen, et al., Comparison of Open-source Cloud Management Platforms: OpenStack and OpenNebula, in *Proceedings of the 9th International Conference on Fuzzy Systems and Knowledge Discovery*, Sichuan, China, 29-31 May, 2012.
[14] OpenStack Official Documentation, August 2017, http://www.openstack.org/foundation/companies/.

زیرنویس‌ها

[1] Infrastructure as a Service
[2] OpenStack
[3] Cloud Computing Center
[4] National Institute of Standards and Technology
[5] Self-service
[6] Platform as a Service
[7] Eucalyptus
[8] platform compatibility
[9] Image
[10] User-friendly
[11] OpenNebula
[12] Application Programming Interface
[13] Apache CloudStack
[14] Minimum Viable Product
[15] Compute
[16] Nova
[17] Instance
[18] Keystone
[19] Swift
[20] Block
[21] Cinder
[22] Neutron
[23] Horizon
[24] Glance
[25] Ceilometer
[26] Heat
[27] Hypervisor
[28] Floating IP
[29] Zabbix
[30] Django

Archive of SID
www.SID.ir