

فصل ۲

مقدمه‌ای بر محاسبات ابری

این فصل مفاهیم اصلی محاسبات ابری را معرفی می‌کند که سرویس‌های پردازشی و ذخیره‌سازی مقیاس‌پذیری را جهت استفاده برای استخراج دانش از مخازن بزرگ داده فراهم می‌کنند. بخش ۲-۱ محاسبات ابری را تعریف کرده و مدل‌های اصلی سرویس و استقراری که توسط فراهم‌کنندگان ابری اتخاذ شده‌اند را مورد بحث قرار می‌دهد. همچنین این بخش تعدادی از بسترهای ابری را توضیح می‌دهد که می‌توانند برای پیاده‌سازی برنامه‌های کاربردی و چارچوب‌هایی برای تحلیل داده‌های توزیع شده مورد استفاده قرار گیرند. بخش ۲-۲ در مورد چگونگی به کارگیری تکنولوژی‌های محاسبات ابری در پیاده‌سازی سیستم‌های تحلیل داده‌های توزیع شده بحث می‌کند. این بخش ابتدا نیازهای اساسی را تعریف می‌کند که باید به وسیله‌ی یک سیستم تحلیل داده‌های توزیع شده پاسخ داده شوند، و سپس به بررسی اینکه چگونه یک بستر ابری می‌تواند برای پاسخ به چنین نیازهایی مورد استفاده قرار گیرد می‌پردازد.

۱-۲ محاسبات ابری: تعریف، مدل‌ها و معماری‌ها

همانطور که در فصل قبل مورد بحث قرار گرفت، یک راه حل موثر برای استخراج دانش مفید از مخازن بزرگ داده‌ها در مدت زمان معقولی به کارگیری روش‌های داده‌کاوی موازی و توزیع شده است. همچنین کار با محیط‌های تحلیل داده که اجازه‌ی دسترسی، مدیریت و داده‌کاوی موثر و کارآمد چنین مخازنی را می‌دهند نیز ضروری و مفید است. به عنوان مثال، یک دانشمند می‌تواند از یک محیط تحلیل داده برای اجرای الگوریتم‌های پیچیده‌ی داده‌کاوی، اعتبار سنجی مدل‌ها، و مقایسه و به اشتراک‌گذاری نتایج با همکارانش در سراسر جهان استفاده کند.

در چند سال اخیر، ابرها به عنوان بسترهای محاسباتی موثری برای روبرو شدن با چالش‌های بازیابی اطلاعات از مخازن بزرگ داده، همچنین فراهم آوردن محیط‌های تحلیل داده‌ی مناسب و کارا برای محققان و شرکت‌ها ظهور کرده‌اند. از دیدگاه یک کاربر، ابر در واقع مفهومی انتزاعی از منابع ذخیره‌سازی و محاسباتی بی‌نهایت مقیاس‌پذیر و دوردست است. از نقطه نظر پیاده‌سازی این دیدگاه، سیستم‌های ابری بر پایه‌ی مجموعه‌ی بزرگی از منابع کامپیوتری شکل گرفته‌اند که در جایی از ابر قرار گرفته‌اند و براساس تقاضا به برنامه‌های کاربردی اختصاص داده می‌شوند (Barga و همکارانش، سال ۲۰۱۱).

بنابراین محاسبات ابری می‌تواند به عنوان یک الگوی توزیع شده تعریف شود که در آن تمام منابع به صورت مقیاس‌پذیر پویا و اغلب مجازی‌سازی شده‌اند و به عنوان سرویس بر روی اینترنت عرضه می‌شوند. مجازی‌سازی روشی نرم‌افزاری است که تفکیک زیرساخت محاسباتی فیزیکی را پیاده‌سازی می‌کند و اجازه می‌دهد منابع مجازی مختلفی روی همان سخت‌افزار ایجاد شود. مجازی‌سازی یک روش پایه‌ای است که محاسبات ابری را قادر می‌سازد تا بر روی یک سرور به صورت همزمان محیط‌های عملیاتی متفاوت و برنامه‌های کاربردی متعددی را اجرا کنند. برخلاف دیگر الگوهای محاسباتی توزیع شده، کاربران در محاسبات ابری نیازی به داشتن دانش و تخصص یا کنترل زیرساختی که در ابر از آنها پشتیبانی می‌کند، ندارند. تعدادی از ویژگی‌هایی که کاربردها، سرویس‌ها، داده‌ها و زیرساخت ابر را تعریف می‌کنند:

- *میزبانی از راه دور*: سرویس‌ها و/یا داده‌ها در زیرساختی که در دوردست قرار دارد میزبانی می‌شوند.
- *همه‌جا حاضر*: سرویس‌ها یا داده‌ها از هر جایی در دسترس هستند.
- *پرداخت به ازای استفاده*: محاسبات ابری، ابزاری قابل اندازه‌گیری است که مشابه ابزارهای سنتی مانند گاز و برق، تنها هزینه‌ی مقدار مصرفی پرداخت می‌شود.

همچنین ما می‌توانیم از تعریف عمومی موسسه‌ی ملی استانداردها و تکنولوژی (NIST¹) از محاسبات ابری برای برجسته نمودن ویژگی‌های اصلی آن استفاده کنیم (Mell و Grance، در سال ۲۰۱۱): "محاسبات ابری مدلی است برای دسترسی راحت و در زمان تقاضا از طریق شبکه به استخری از منابع قابل پیکربندی و به اشتراک گذاشته شده (به عنوان مثال شبکه‌ها، سرورها، مخازن، برنامه‌های کاربردی و خدمات) که به سرعت و با کمترین تلاش مدیریتی یا دخالت فراهم‌کننده می‌توانند تهیه و توزیع شوند". بر اساس تعریف ارائه شده توسط NIST، می‌توانیم پنج ویژگی اساسی سیستم‌های محاسبات ابری را به صورت در زمان تقاضا، خود-سرویس، دسترسی گسترده‌ی شبکه‌ای، استخری از منابع، انعطاف‌پذیری سریع و سرویس قابل‌اندازه‌گیری تعریف کنیم.

سیستم‌های ابری می‌توانند بر اساس مدل‌های سرویس^۲ (نرم‌افزار به عنوان سرویس^۳، بستر به عنوان سرویس^۴ و زیرساخت به عنوان سرویس^۵) و مدل‌های استقرار^۶ (ابر عمومی^۷، ابر خصوصی^۸ و ابر آمیخته^۱) طبقه‌بندی شوند.

¹ National Institute of Standards and Technology (NIST)

² Service Models

³ Software as a Service

⁴ Platform as a Service

⁵ Infrastructure as a Service

⁶ Deployment Models

⁷ Public Cloud

⁸ Private Cloud

۲-۱-۱ مدل‌های سرویس در محاسبات ابری

فروشنندگان محاسبات ابری سرویس‌های خود را بر طبق سه مدل اصلی فراهم می‌کنند: نرم‌افزار به عنوان سرویس (SaaS)، بستر به عنوان سرویس (PaaS) و زیرساخت به عنوان سرویس (IaaS).

نرم‌افزار به عنوان سرویس، یک مدل تحویل تعریف می‌کند که در آن نرم‌افزار و داده‌ها به عنوان سرویس‌های آماده برای استفاده از طریق اینترنت در اختیار مشتری قرار می‌گیرند. نرم‌افزار و داده‌های مرتبط بوسیله‌ی فراهم‌کنندگان میزبانی می‌شوند و مشتری‌ها بدون نیاز به سخت‌افزار یا نرم‌افزار اضافی می‌توانند به آنها دسترسی داشته باشند. علاوه بر این، مشتری‌ها در حالت عادی بدون نیاز به خرید زیرساخت یا حق مالکیت نرم‌افزاری اضافی، هزینه‌ی ماهیانه/سالانه‌ای را پرداخت می‌کنند. از سیستم‌های Webmail (مانند Gmail)، تقویم‌ها (تقویم یا هو^۲)، مدیریت سند^۳ (Microsoft Office 365)، دستکاری در عکس (Photoshop Express)، مدیریت ارتباط مشتری (Salesforce) و غیره نمونه‌های رایجی از کاربردهای SaaS هستند.

در مدل بستر به عنوان سرویس، فروشنندگان ابر یک بستر محاسباتی را که معمولاً شامل پایگاه‌داده‌ها، سرویس‌دهندگان برنامه‌های کاربردی و محیط توسعه است را برای ایجاد، آزمایش و اجرای برنامه‌های کاربردی به مشتریان عرضه می‌کنند. توسعه‌دهندگان می‌توانند تنها بر روی استقرار برنامه‌های کاربردی تمرکز کنند چرا که فراهم‌کنندگان ابر مسئول نگهداری و بهینه‌سازی محیط و زیرساخت هستند. از این رو، مشتریان در راستای توسعه‌ی برنامه‌های کاربردی از مجموعه‌ای از سرویس‌های محیطی استفاده می‌کنند که ماجولار هستند و می‌توانند براحتی با یکدیگر یکپارچه شوند. به طور معمول، برنامه‌های کاربردی به عنوان SaaS و آماده‌ی استفاده توسعه داده می‌شوند. Google App Engine، Microsoft Azure و Salesforce.com نمونه‌هایی از محیط‌های ابری PaaS هستند.

در نهایت، زیرساخت به عنوان سرویس یک مدل برون‌سپاری^۴ است که در آن مشتریان منابعی مانند CPU، دیسک‌ها یا منابع پیچیده‌تر مانند سرورهای مجازی شده یا سیستم‌های عامل را برای پشتیبانی از عملیاتشان اجاره می‌کنند (به عنوان مثال Amazon EC2 و RackSpaceCloud). معمولاً کاربران مدل IaaS دارای مهارت‌های سیستمی و مدیریت شبکه می‌باشند، چنانکه آنها باید با پیکربندی، عملیات و نگهداری وظایف سروکار داشته باشند. در مقایسه با رویکرد

¹ Hybrid Cloud

² Yahoo Calendar

³ Document management

⁴ Outsourcing

IaaS، **PaaS**، مدل **IaaS** هزینه‌های مدیریت سیستمی بالایی را برای کاربران دارد؛ از سوی دیگر **IaaS** قابلیت سفارشی‌سازی محیط اجرایی را به طور کامل امکان می‌سازد. توسعه‌دهندگان می‌توانند با افزودن یا کاستن ماشین‌های مجازی در مقیاس سرویس‌های خود تغییر ایجاد کنند.

در جدول ۱-۲، سه مدل سرویس از لحاظ انعطاف‌پذیری، مقیاس‌پذیری، قابلیت حمل، امنیت، نگهداری و هزینه با یکدیگر مقایسه شده است.

جدول ۱-۲: چگونه IaaS ، PaaS ، SaaS و نیازهای توسعه‌دهندگان و کاربران نهایی را برآورده می‌کنند.			
<i>IaaS</i>	<i>PaaS</i>	<i>SaaS</i>	نیازها
توسعه‌دهندگان مجبورند سرورهایی را که قرار است برنامه‌های کاربردی آنها را میزبانی کنند، ایجاد نموده و پیکربندی سیستم‌عامل و ماژول‌های نرم‌افزاری که قرار است بر روی این سرورها اجرا شوند را خود برعهده بگیرند.	توسعه‌دهندگان با استفاده از کتابخانه‌ها و ابزار پشتیبانی سازگار با بستر، برنامه‌های کاربردی خود را نوشته، سفارشی‌سازی نموده و آزمایش می‌کنند. کاربران می‌توانند نوع منابع محاسباتی و ذخیره‌سازی مجازی را انتخاب کنند که برای اجرای برنامه‌های کاربردی آنها مورد استفاده قرار می‌گیرند.	کاربران می‌توانند واسط برنامه‌های کاربردی را شخصی‌سازی کرده و رفتار آن را کنترل کنند ولی نمی‌توانند تصمیم بگیرند که چه اجزای سخت‌افزاری و نرم‌افزاری برای پشتیبانی از اجرای این برنامه‌ها مورد استفاده قرار گیرند.	انعطاف‌پذیری
توسعه‌دهندگان می‌توانند منابع محاسباتی و ذخیره‌سازی را به طور اتوماتیک مقیاس می‌پذیرند. استفاده کنند ولی برنامه‌های کاربردی آنها باید	مشابه مدل SaaS ، منابع محاسباتی و ذخیره‌سازی زیربنا معمولاً به طور اتوماتیک مقیاس می‌پذیرند.	منابع محاسباتی و ذخیره‌سازی زیربنا، معمولاً به طور اتوماتیک برای مطابقت با برنامه‌ی کاربردی مورد تقاضا	مقیاس‌پذیری

<p>مقیاس پذیر باشد و اجازهی استفادهی پویا از منابع جدید را بدهد.</p>		<p>مقیاس می یابند، بنابراین کاربران مجبور نیستند منابع را دستی اختصاص دهند. نتیجه تنها به درجهی ارتجاعی که سیستم ابر فراهم می کند بستگی دارد.</p>	
<p>در صورتی که یک فراهم کننده اجازهی دانلود یک ماشین مجازی را در فرمت استاندارد می دهد، در این صورت ممکن است که بتوان در این مدل به یک فراهم کنندهی دیگر انتقال یافت.</p>	<p>برنامه های کاربردی تنها در صورتی می توانند به فراهم کنندهی دیگری انتقال یابند که فراهم کنندهی جدید، سرویس ها و ابزارهای بستر مورد نیاز را با فراهم کنندهی قبلی به اشتراک بگذارد.</p>	<p>در اینجا انتقال برنامه های کاربردی به دیگر فراهم کنندگان ابر می تواند مشکلاتی را ایجاد کند به این دلیل که تعدادی از نرم افزارها و ابزارها بر روی سیستم های مختلف نمی توانند کار کنند. به عنوان مثال، داده های برنامه ی کاربردی ممکن است در فرمتی باشد که فراهم کنندهی دیگر نتواند آنرا بخواند.</p>	<p>قابلیت حمل</p>
<p>توسعه دهندگان باید خود مراقب مسائل امنیتی در همه ی قسمت ها، از سیستم عامل گرفته تا لایه های برنامه های</p>	<p>مسئولیت امنیت کدها و کتابخانه هایی که برای ایجاد برنامه های کاربردی استفاده شده اند بر عهده ی خود توسعه دهنده می باشد.</p>	<p>کاربران می توانند تنها تعدادی از تنظیمات امنیتی برنامه های کاربردی خود را کنترل کنند (به عنوان مثال، استفاده از https به جای</p>	<p>امنیت</p>

<p>کاربردی باشند.</p>		<p>http در دسترسی به برخی صفحات وب). لایه‌های امنیتی اضافی (به عنوان مثال، تکرار داده‌ها) از کاربران مخفی است و مستقیماً بوسیله سیستم مدیریت می‌شوند.</p>	
<p>توسعه‌دهندگان مسئول نگهداری از تمام اجزای نرم‌افزاری هستند که شامل سیستم عامل نیز می‌شود؛ سخت افزار بوسیلهی فراهم کننده نگهداری می‌شود.</p>	<p>توسعه‌دهندگان فقط مسئول نگهداری از برنامه‌های کاربردی خود هستند؛ دیگر اجزای نرم‌افزاری و سخت‌افزاری بوسیلهی فراهم کننده نگهداری می‌شوند.</p>	<p>کاربران مجبور نیستند وظایف نگهداری را بر عهده بگیرند.</p>	<p>نگهداری</p>
<p>توسعه‌دهندگان برای همه‌ی منابعی نرم‌افزاری و سخت‌افزاری که استفاده می‌کنند هزینه پرداخت می‌کنند.</p>	<p>توسعه‌دهندگان برای منابع محاسباتی و ذخیره‌سازی، همچنین مجوز کتابخانه‌ها و ابزاری که توسط برنامه‌های کاربردی آنها مورد استفاده قرار می‌گیرند، هزینه پرداخت می‌کنند.</p>	<p>کاربران معمولاً هزینه‌ی ماهیانه/سالانه‌ای را در قبال استفاده از نرم‌افزارها پرداخت می‌کنند و هیچ هزینه‌ی اضافی را برای زیرساخت نمی‌پردازند.</p>	<p>هزینه</p>

۲-۱-۲ مدل‌های استقرار در محاسبات ابری

سرویس‌های محاسبات ابری مطابق سه مدل استقرار اصلی: عمومی، خصوصی و آمیخته^۱ ارائه می‌شوند.

یک فراهم‌کننده‌ی ابر عمومی، سرویس‌های خود را به عموم مردم از طریق اینترنت عرضه می‌کند. کاربران یک ابر عمومی، کنترل کمی بر روی تکنولوژی زیرساخت دارند و یا فاقد چنین کنترلی هستند. در این مدل، سرویس‌ها می‌توانند به صورت رایگان و یا بر طبق سیاست پرداخت به ازای مصرف ارائه شوند. فراهم‌کنندگان عمومی اصلی از قبیل **Google**، **Microsoft** و **Amazon** دارای مراکز داده‌ی^۲ اختصاصی بوده و مدیریت و عرضه‌ی سرویس‌های خود را بر روی این مراکز انجام می‌دهند.

یک فراهم‌کننده‌ی ابر خصوصی، عملیات و قابلیت‌هایی را به عنوان سرویس عرضه می‌کند که بر روی شبکه‌ی اینترنت یک شرکت یا در یک مرکز داده‌ی دور دست میزبانی می‌شوند. به دلیل راه‌حل‌های امنیتی پیشرفته و کنترل داده‌ای که مدل ابر خصوصی ارائه می‌دهد و این راه‌حل‌ها در مدل ابر عمومی وجود ندارد اغلب شرکت‌های **IT** کوچک و متوسط مدل ابر خصوصی را ترجیح می‌دهند.

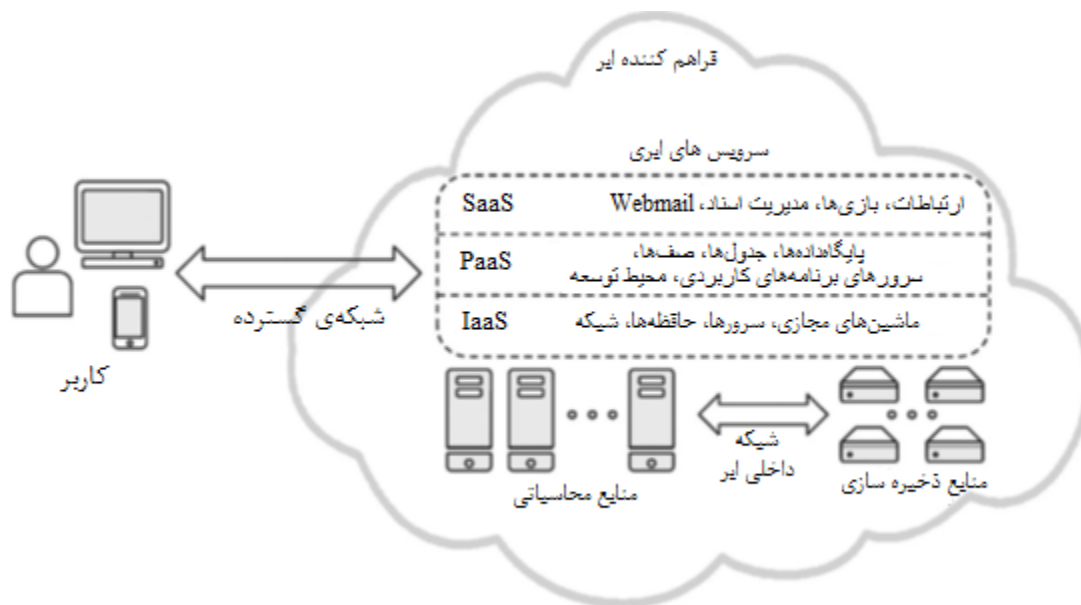
در نهایت، یک ابر آمیخته در واقع ترکیبی از دو یا چندین ابر (عمومی یا خصوصی) است که اجزاء مختلف باقی می‌مانند ولی به یکدیگر وصل شده‌اند. شرکت‌ها می‌توانند ابرهای اختصاصی خود را با استفاده از ابرهای خصوصی شرکت‌های همکار یا ابرهای عمومی گسترش دهند. به ویژه اینکه، با گسترش زیرساخت خصوصی با منابع ابر عمومی، سرویس دادن به بیشترین درخواست‌ها، سرویس‌دهی بهتر به درخواست کاربران و پیاده‌سازی استراتژی‌های با حداکثر قابلیت دسترسی را ممکن می‌سازد.

شکل ۱-۲ معماری کلی یک ابر عمومی و اجزای اصلی آن را به تصویر می‌کشد (**Li** و همکارانش، در سال ۲۰۱۰). سرویس‌های محاسبات ابری با استفاده از تجهیزات کاربران از قبیل کامپیوترهای رومیزی، لپ‌تاپ‌ها و تلفن‌های هوشمند ارائه می‌شوند. کاربران از طریق این تجهیزات و با استفاده از مرورگرها یا برنامه‌های کاربردی رومیزی/همراه می‌توانند به سرویس‌های مبتنی بر ابر دسترسی و با آن‌ها تعامل داشته باشند. نرم‌افزار تجاری و داده‌های کاربران بر روی سرورهایی اجرا و ذخیره می‌شوند که در مراکز داده‌ی ابری میزبانی شده و منابع محاسباتی و ذخائر را فراهم می‌کنند. منابع شامل هزاران

¹ Hybrid

² Data Centers

سرور و وسایل ذخیره‌سازی هستند که از طریق شبکه‌ی داخلی ابر^۱ به یکدیگر متصل شده‌اند. تبادل داده بین کاربران و مراکز داده با استفاده از شبکه‌ی گسترده^۲ است.



شکل ۱-۲: معماری کلی یک ابر عمومی.

تکنولوژی‌ها و استانداردهای متعددی می‌توانند بوسیله‌ی اجزای مختلف موجود در معماری مورد استفاده قرار گیرند. به عنوان مثال، کاربران می‌توانند با سرویس‌های ابر از طریق وب سرویس‌های برپایه‌ی SOAP یا دیگر وب سرویس‌ها تعامل داشته باشند (Richardson و Ruby، در سال ۲۰۰۷). تکنولوژی‌های HTML5 و Ajax به واسطه‌های وب اجازه می‌دهند تا دید و تعامل معادلی با سرویس‌های ابری مانند دیگر نرم‌افزارهای کاربردی رومیزی داشته باشند. واسطه محاسباتی ابری باز^۳ (OCCI) مشخص می‌کند که فراهم‌کنندگان ابر چگونه می‌توانند منابع محاسباتی، داده و شبکه‌ای خود را از طریق واسطه‌های استاندارد عرضه کنند. نمونه‌ی دیگر فرمت مجازی‌سازی باز^۴ (OVF) است که برای بسته‌بندی و توزیع وسایل یا نرم‌افزارهای مجازی (به عنوان مثال سیستم عامل‌های مجازی) برای اجرا شدن در ماشین‌های مجازی است.

¹ Intracloud Network

² Wide-area Network

³ Open Cloud Computing Interface

⁴ Open Virtualization Format

۲-۱-۳ محیط‌های ابری

در این بخش چهار نمونه به عنوان نماینده‌ای از محیط‌های ابری معرفی می‌شود که **Microsoft Azure** به عنوان نمونه‌ای از **PaaS** عمومی، **Amazon Web Service** به عنوان مشهورترین **IaaS** عمومی، **OpenNebula** و **OpenStack** به عنوان نمونه‌هایی از **IaaS** خصوصی هستند. این محیط‌ها می‌توانند برای پیاده‌سازی برنامه‌های کاربردی و چارچوب‌های کاری برای تحلیل داده‌ها در ابر مورد استفاده قرار گیرند.

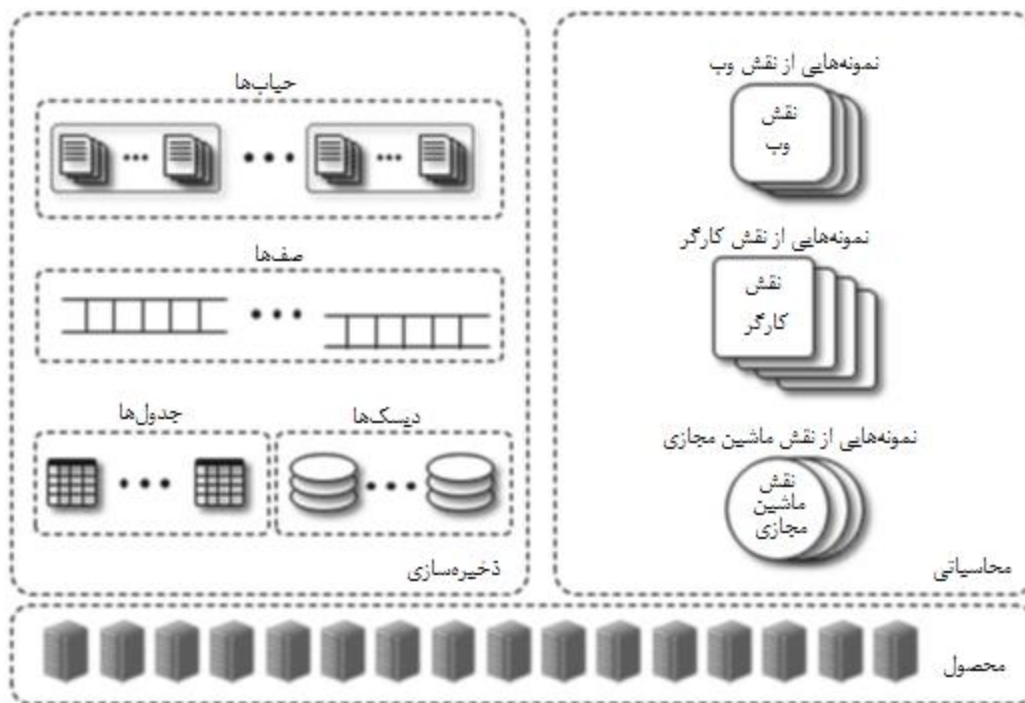
۲-۱-۳-۱ Microsoft Azure

Azure یک محیط و مجموعه‌ای از سرویس‌های ابری است که می‌تواند برای توسعه‌ی برنامه‌های کاربردی مبتنی بر ابر یا برای ارتقای برنامه‌های کاربردی موجود با قابلیت‌های مبتنی بر ابر مورد استفاده قرار گیرد. منابع محاسباتی و ذخیره‌سازی مبتنی بر تقاضا که این بستر فراهم می‌کند با استفاده از به کارگیری قدرت محاسباتی و ذخیره‌سازی مراکز داده‌ی **Microsoft Azure** برای پشتیبانی از دسترس‌پذیری بالا و سرویس‌های با مقیاس‌پذیری پویا طراحی شده است که با مدلی که کاربران در آن هزینه‌ای به ازای آنچه استفاده نموده‌اند پرداخت می‌کنند، تطابقت دارد.

بستر **Azure** برای ذخیره‌سازی پایگاه‌داده‌های بزرگ، اجرای حجم بزرگی از محاسبات اجرایی و توسعه‌ی برنامه‌های کاربردی **SaaS** که کاربران نهایی را هدف قرار می‌دهد، می‌تواند مورد استفاده قرار گیرد.

Microsoft Azure شامل سه سرویس/قسمت است که در شکل ۲-۲ نشان داده شده است.

- قسمت محاسبه: محیط محاسباتی که برای اجرای برنامه‌های کاربردی ابری است. هر برنامه‌ی کاربردی بر اساس نقش‌ها ساخت یافته است: نقش وبی، برای برنامه‌های کاربردی مبتنی بر وب؛ نقش کارگر، برای برنامه‌های کاربردی اجرایی؛ نقش ماشین مجازی، برای **image**های ماشین مجازی.
- قسمت ذخیره: منابع ذخیره‌سازی مقیاس‌پذیر فراهم می‌کند که برای مدیریت داده‌های متنی و باینری (**Blobs**)، جدول‌های غیررابطه‌ای (**Tables**)، صف‌هایی که برای ارتباط ناهمزمان بین اجزاء (**Queues**) و دیسک‌های مجازی (**Disks**).
- قسمت کنترل‌کننده‌ی اجزاء: هدف این قسمت ایجاد شبکه‌ای بهم متصل از نودها در ماشین‌های فیزیکی یک مرکز داده است. سرویس‌های محاسباتی و ذخیره‌سازی بر روی این جزء ساخته می‌شوند.



شکل ۲-۲: Microsoft Azure.

Microsoft Azure واسط‌های استاندارد را فراهم می‌کند که به توسعه‌دهندگان اجازه‌ی تعامل با سرویس‌هایش را می‌دهد. علاوه بر این، توسعه‌دهندگان می‌توانند از محیط‌های توسعه‌ی مجتمع^۱ (IDE) مانند Microsoft Visual Studio و Eclipse برای طراحی و انتشار آسانتر برنامه‌های کاربردی Azure استفاده کنند.

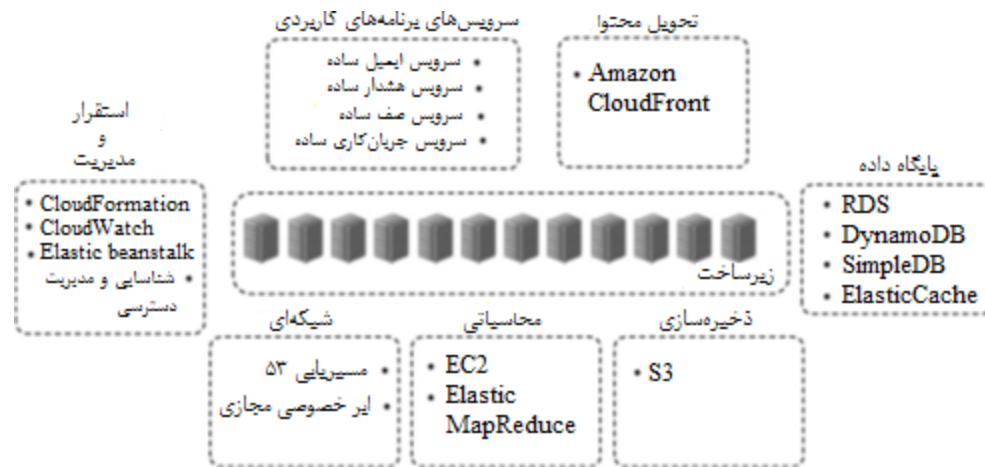
۲-۱-۳-۲ سرویس‌های وب آمازون (Amazon Web Services)

Amazon منابع محاسباتی و ذخیره‌سازی زیرساخت IT خود را به شکل سرویس‌های وب در اختیار توسعه‌دهندگان قرار می‌دهد. سرویس‌های وب Amazon^۲ (AWS) در واقع مجموعه‌ی بزرگی از سرویس‌های ابری است که می‌توانند بوسیله‌ی کاربران برای ایجاد برنامه‌های کاربردی SaaS یا ادغام نرم‌افزارهای معمولی با قابلیت‌های ابری تشکیل شوند (شکل ۲-۳). از آنجایی که Amazon^۳ SDKهایی را برای اهداف برنامه‌نویسی با زبان‌ها و بسترهای مختلف (به عنوان مثال Java، .Net، PHP و Android) فراهم کرده است، تعامل با سرویس‌های Amazon ساده است.

¹ Integrated Development Environment

² Amazon Web Services

³ Software development kit



شکل ۲-۳: Amazon Web Services

AWS شامل سرویس‌های اصلی زیر است:

- محاسباتی: ابر محاسباتی ارتجاعی (EC2^۱) اجازه‌ی ایجاد و اجرای سرورهای مجازی را می‌دهد؛ Amazon Elastic MapReduce برای ایجاد و اجرای برنامه‌های کاربردی MapReduce هستند.
- ذخیره‌سازی: سرویس ذخیره‌سازی ساده^۲ (S3) که اجازه‌ی ذخیره‌سازی و بازیابی داده‌ها را از طریق اینترنت می‌دهد.
- پایگاه‌داده: سرویس پایگاه داده‌ی رابطه‌ای^۳ (RDS) برای جدول‌های رابطه‌ای؛ DynamoDB برای جدول‌های غیررابطه‌ای؛ SimpleDB برای مدیریت پایگاه‌داده‌های کوچک؛ کش ارتجاعی^۴ برای کش نمودن داده‌ها.
- شبکه: Route 53 که یک DNS Web Service است؛ ابر خصوصی مجازی که برای پیاده‌سازی یک شبکه مجازی می‌باشد.
- استقرار و مدیریت: CloudFormation برای ایجاد مجموعه‌ای از ماشین‌های مجازی آماده‌ی استفاده به همراه نرم‌افزارهای از پیش نصب شده (به عنوان مثال برنامه‌های کاربردی وب)؛ CloudWatch برای نظارت بر منابع AWS؛ Elastic Beanstalk برای استقرار و اجرای برنامه‌های کاربردی مشتری که به زبان‌های Java، PHP و دیگر زبان‌ها نوشته شده‌اند؛ مدیریت هویت و دسترسی برای کنترل امنیتی دسترسی به منابع و سرویس‌های AWS.
- تحویل محتوا: Amazon CloudFront توزیع محتوا را از طریق شبکه‌ی عمومی آسان می‌سازد.

¹ Elastic Compute Cloud

² Simple Storage Service

³ Relational Database Services

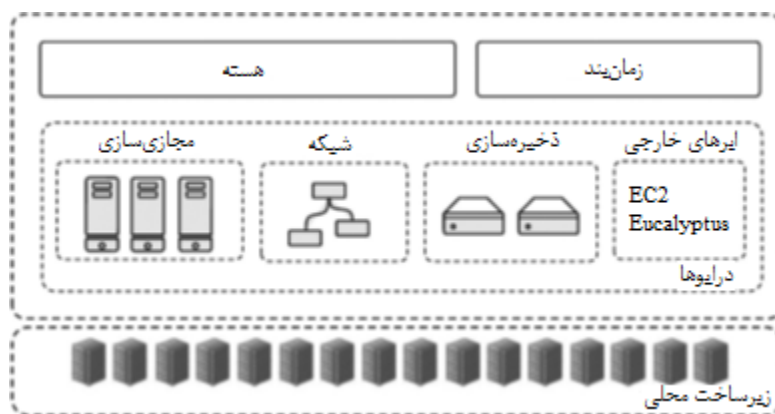
⁴ Elastic Cache

- سرویس‌های برنامه‌های کاربردی: سرویس پست الکترونیکی ساده که یک سرویس ارسال پست الکترونیک پایه‌ای را فراهم می‌کند. سرویس اطلاع‌رسانی ساده که برای اعلام به کاربران است؛ سرویس صف ساده که صفی از پیام‌ها را پیاده‌سازی می‌کند؛ سرویس گردش کار ساده که برای پیاده‌سازی برنامه‌های کاربردی مبتنی بر گردش کاری است.

با وجود اینکه Amazon به عنوان اولین فراهم‌کننده‌ی IaaS شناخته می‌شود (بر اساس سرویس‌های مبتنی بر EC2 و S3 آن)، ولی امروزه با سرویس‌هایی مانند Elastic Beanstalk به عنوان فراهم‌کننده‌ی PaaS نیز فعالیت می‌کند.

۳-۳-۱-۲ OpenNebula

OpenNebula (Sotomayor و همکارانش، در سال ۲۰۰۹) در واقع یک چارچوب اصلی متن‌باز است که برای ایجاد ابرهای خصوصی و آمیخته مورد استفاده قرار می‌گیرد. جزء اصلی معماری OpenNebula، هسته است که ماشین‌های مجازی را بوسیله‌ی اتصال آنها با یک محیط شبکه‌ی مجازی ایجاد و کنترل می‌کند (شکل ۲-۴ را ببینید). علاوه بر این، هسته برای انجام عملیات ذخیره‌سازی، شبکه و مجازی‌سازی با اجزاء قابل اتصالی^۱ که درایور نامیده می‌شوند، تعامل دارد. به این ترتیب، OpenNebula از زیرساخت اساسی مستقل است و یک محیط یکنواخت مدیریتی را عرضه می‌کند.



شکل ۲-۴: معماری OpenNebula.

همچنین هسته از استقرار سرویس‌ها نیز پشتیبانی می‌کند که مجموعه‌ای از اجزای بهم متصل (به عنوان مثال وب سرور، پایگاه داده) و در حال اجرا بر روی ماشین‌های مجازی متعددی هستند. جزء دیگر زمان‌بند است که مسئول تخصیص

¹ Pluggable

ماشین‌های مجازی بر روی سرورهای فیزیکی است. برای این منظور، زمان‌بند از طریق دستورات تعبیه شده‌ی مناسبی با هسته تعامل دارد.

OpenNebula می‌تواند با استفاده از درایورهای مخصوصی به عنوان یک ابر آمیخته پیاده‌سازی شود که با ابرهای بیرونی تعامل داشته باشد. به این ترتیب، زیرساخت محلی می‌تواند با منابع محاسباتی و ذخیره‌سازی ابرهای عمومی الحاق شود. در حال حاضر، **OpenNebula** شامل درایورهایی برای استفاده از منابع **Amazon EC2** و دیگر چارچوب کاری متن‌باز به نام **Eucalyptus** است (Nurmi و همکارانش، در سال ۲۰۰۹).

۲-۱-۳-۴ OpenStack

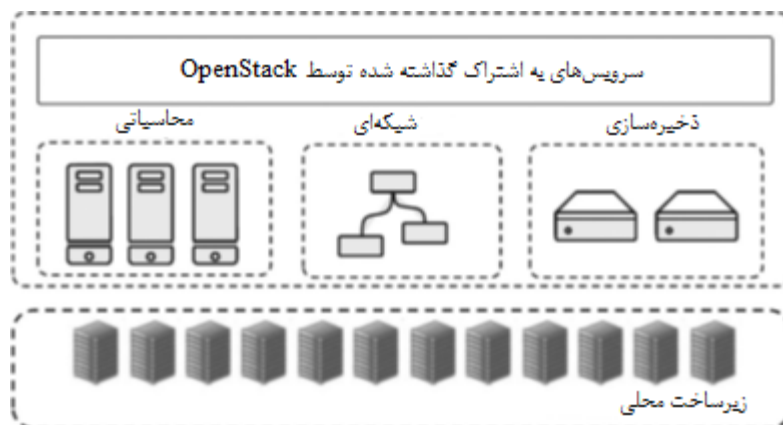
OpenStack^۱ در واقع یک سیستم‌عامل ابری است که مدیریت مجموعه‌ی بزرگی از منابع پردازشی، ذخیره‌سازی و شبکه‌ای را در یک مرکز داده از طریق واسط‌های مبتنی بر وب فراهم می‌آورد. سیستم موردنظر برای چهار هدف اصلی طراحی، توسعه و توزیع شده است:

- منبع باز: **OpenStack** تحت قوانین **Apache** منتشر شده است.
- طراحی باز: هر شش ماه، یک نشست طراحی برای جمع‌آوری نیازمندی‌ها و تعریف مشخصات فنی جدید برای نسخه‌های بعدی برگزار می‌شود.
- توسعه‌ی باز: مخزنی از منابع کد در دسترس عموم قرار دارد که برای مراحل توسعه نگهداری می‌شود.
- جامعه‌ی باز: اکثر تصمیم‌ها توسط جامعه‌ی **OpenStack** و با استفاده از مدل کنفرانس تنبل^۲ اتخاذ می‌شود.

معماری ماژولار **OpenStack** بوسیله‌ی چهار جزء اصلی تشکیل شده است که در شکل ۲-۵ نشان داده شده است.

^۱ OpenStack, <http://www.openstack.org/>

^۲ Lazy Consensus Model



شکل ۲-۵: OpenStack.

قسمت محاسبه، سرورهای مجازی بر حسب تقاضا را با مدیریت منابع پردازشی موجود در مرکز داده فراهم می‌کند. تکنولوژی‌های مختلفی می‌تواند توسط این سیستم پشتیبانی شود (به عنوان مثال VMware، KVM) و به صورت افقی می‌تواند مقیاس یابد. قسمت ذخیره‌سازی در OpenStack، یک سیستم ذخیره‌سازی مقیاس‌پذیر و اضافی را فراهم می‌کند. این قسمت ذخیره‌سازی اشیاء و بلوک را پشتیبانی می‌کند که ذخیره‌سازی اشیاء اجازه‌ی ذخیره‌سازی و بازیابی اشیاء و فایل‌ها را در مرکز داده می‌دهد و ذخیره‌سازی بلوک امکان ایجاد، اتصال و جداسازی تجهیزات بلوکی سرورها را می‌دهد. قسمت شبکه در OpenStack، شبکه‌ها و آدرس‌های IP را مدیریت می‌کند. در نهایت، سرویس‌های به اشتراک گذاشته شده در OpenStack، سرویس‌های اضافی هستند که برای راحتی استفاده از مرکز داده فراهم شده‌اند. به عنوان مثال، سرویس هویت برای این است که کاربران و سرورها را بهم نگاشت کند، سرویس تصویری برای مدیریت سرورهای تصاویر و سرویس پایگاه‌داده نیز یک پایگاه‌داده‌ی رابطه‌ای را فراهم می‌کند.

۲-۲ سیستم‌های محاسبات ابری برای برنامه‌های کاربردی

مبتنی بر داده

سیستم‌های ابری می‌توانند به طور موثری برای پشتیبانی از برنامه‌های کاربردی داده‌های متمرکز به کار گرفته شود چرا که آنها ذخائر انعطاف‌پذیر و سرویس‌های پردازشی را به خوبی بسترهای نرم‌افزاری برای توسعه دادن و اجرای محیط‌های تحلیل داده را بر فراز چنین سرویس‌هایی فراهم می‌کنند. در این قسمت در مورد تکنولوژی‌های ابری بحث می‌شود که

می‌توانند جهت پیاده‌سازی سیستم‌های تحلیل داده برای برنامه‌های کاربردی داده‌های متمرکز **KDD** به کار گرفته شود. ابتدا تعریف نیازهای کاربردی و غیر کاربردی مشخص می‌شود که باید هر برنامه‌ی کاربردی **KDD** که یک سیستم تحلیل داده‌ی توزیع شده باید این نیازها را برآورده کند. نیازهای کاربردی مشخص می‌کنند که کدام ویژگی‌های سیستم باید تامین شود و نیازهای غیر کاربردی به معیارهای کیفی اشاره دارد که بیشتر به کارایی سیستم مربوط می‌شود.

۲-۲-۱ نیازهای کاربردی

نیازهای کاربردی که توسط یک سیستم تحلیل داده‌ی توزیع شده باید برآورده شود به دو دسته‌ی اصلی تقسیم می‌شود: نیازهای مدیریت منابع و نیازهای مدیریت برنامه‌های کاربردی. نیازهای مدیریت منابع به نیازهایی اشاره دارد که به مدیریت همه‌ی منابع مربوط می‌شود (داده‌ها، ابزار، نتایج) که ممکن است یک برنامه‌ی کاربردی **KDD** را نیز شامل شود؛ نیازهای مدیریت برنامه‌های کاربردی به طراحی و اجرای خود برنامه‌های کاربردی مربوط می‌شود.

۲-۲-۱-۱ مدیریت منابع

منابعی که مورد توجه برنامه‌های کاربردی **KDD** هستند شامل منابع داده، ابزار کشف دانش و نتایج کشف دانش می‌باشند. بنابراین، یک سیستم تحلیل داده‌ی توزیع شده باید به نیازهای مدیریت منابعی که در ادامه ذکر می‌شوند بپردازد:

- مدیریت داده: منابع داده می‌توانند در قالب‌های متفاوتی از قبیل پایگاه‌های داده‌ی رابطه‌ای، فایل‌های ساده، یا سندهای نیمه‌ساختار یافته (به عنوان مثال فایل‌های **XML**) باشند. سیستم باید روش‌هایی را برای ذخیره کردن و دسترسی به چنین منابع داده‌ای را مستقل از قالب مخصوصشان فراهم آورد. به علاوه، ابرداده‌ها باید به صورت رسمی و فرمولی تعریف شوند و باید توصیف اطلاعات مرتبط در رابطه با منابع داده مورد استفاده قرار گیرد (به عنوان مثال مکان قرارگیری، قالب، دسترس‌پذیری، نمایش در دسترس) تا بتوان دسترسی و به کارگیری موثر آنها را قادر ساخت.
- مدیریت ابزار: ابزار کشف دانش که شامل الگوریتم‌ها و سرویس‌ها برای انتخاب داده، پیش پردازش، تبدیل، داده‌کاوی و ارزیابی نتایج است. سیستم باید روش‌هایی را برای دسترسی و استفاده‌ی چنین ابزاری مستقل از

پیاده‌سازی مخصوصشان را فراهم آورد. ابر داده باید برای توصیف مهمترین ویژگی‌های ابزار KDD (مانند عملکردشان، مکان قرارگیری، نحوه استفاده) مورد استفاده قرار گیرد.

- مدیریت نتیجه: دانش به دست آمده از نتایج مرحله‌ی کشف دانش به وسیله‌ی یک مدل دانش (یا مدل داده‌کاوی) نشان داده می‌شود. سیستم باید روش‌هایی را برای ذخیره‌سازی و دسترسی چنین مدل‌هایی مستقل از ساختار و قالبشان فراهم آورد. مانند داده و ابزار، مدل‌های داده‌کاوی نیز نیاز دارند که محتوایشان به وسیله‌ی ابر داده توضیح و تفسیر شوند تا بازیابی موثرشان را قادر سازد.

۲-۱-۲-۲ مدیریت برنامه‌های کاربردی

یک سیستم تحلیل داده‌ی توزیع شده باید روش‌های موثری را برای طراحی برنامه‌های کاربردی داده‌های متمرکز KDD (مدیریت طراحی) و کنترل اجرای آنها (مدیریت اجرا) را فراهم آورد.

- مدیریت طراحی: تمام محدوده‌ی برنامه‌های کاربردی تحلیل داده‌ی توزیع شده از وظایف ساده‌ی داده‌کاوی گرفته تا الگوهای پیچیده‌ی داده‌کاوی با عنوان گردش‌های کاری^۱ بیان می‌شوند. از دید طراحی، سه دسته‌ی اصلی برنامه‌های کاربردی کشف دانش می‌تواند تعریف شود: *برنامه‌های کاربردی تک وظیفه‌ای*، که در آن کشف مانند دسته‌بندی، خوشه‌بندی یا قوانین انجمنی بر روی یک منبع داده انجام می‌شود؛ *برنامه‌های کاربردی پارامتر گسترده*، که در آن یک مجموعه داده با استفاده از نمونه‌های متعددی از یک الگوریتم داده‌کاوی یکسان ولی با پارامترهای مختلف مورد تحلیل و بررسی قرار می‌گیرد. *برنامه‌های کاربردی بر اساس گردش کاری*، که در آن برنامه‌های کاربردی کشف دانش احتمالاً پیچیده‌ای با گراف‌هایی که منابع داده را به یکدیگر متصل می‌کنند، الگوریتم‌های داده‌کاوی و ابزارهای تصویرسازی و بصری مشخص می‌شوند. یک سیستم کلی باید محیط‌هایی را برای طراحی موثر همه‌ی دسته‌های فوق‌الذکر در برنامه‌های کاربردی تحلیل داده فراهم کند.

- مدیریت اجرا: سیستم باید یک محیط اجرای توزیع شده را فراهم آورد که اجرای موثر برنامه‌های کاربردی تحلیل داده را پشتیبانی کند که به وسیله‌ی کاربران طراحی شده‌اند. از آنجایی که محدوده‌ی برنامه‌های کاربردی از تک وظیفه‌ای گرفته تا گردش‌های کاری کشف دانش پیچیده است بنابراین محیط اجرایی باید از عهده‌ی اجرای چنین تنوعی در برنامه‌های کاربردی را داشته باشد. به ویژه، محیط اجرایی باید قابلیت‌های زیر را فراهم کند که به

¹ Workflows

مراحل مختلف اجرای یک برنامه‌ی کاربردی مربوط است: دسترسی به منابع داده‌ای که مورد داده‌کاوی قرار گرفته است، اختصاص دادن منابع محاسباتی مورد نیاز این برنامه‌ها، اجرای برنامه‌ها بر اساس مشخصاتی که کاربر تعیین کرده است، که ممکن است به عنوان یک گردش کاری بیان شود؛ نتایج را به کاربر نمایش دهد. همچنین، سیستم باید به کاربران اجازه‌ی نظارت بر اجرای برنامه‌های کاربردی را بدهد.

۲-۲-۲ نیازهای غیر کاربردی

نیازهای غیر کاربردی می‌توانند در سه سطح تعریف شوند: کاربر، معماری و زیرساخت. نیازهای کاربر مشخص می‌کنند که کاربر چگونه باید با سیستم تعامل داشته باشد؛ نیازهای معماری مشخص می‌کنند که کدام قواعد و قوانین در الهام بخشیدن طراحی معماری سیستم مورد استفاده قرار می‌گیرند؛ در نهایت، نیازهای زیرساختی ویژگی‌های غیرکارکردی را در زیرساخت محاسباتی توصیف می‌کنند.

۲-۲-۲-۱ نیازهای کاربر

از نقطه نظر یک کاربر، نیازهای غیر کاربردی زیر باید برآورده شود:

- قابلیت استفاده: سیستم باید به سادگی توسط کاربران انتهایی مورد استفاده قرار گیرد بدون اینکه نیاز به سپری کردن دوره‌های آموزشی خاصی داشته باشند.
- دسترسی از همه جا: کاربران باید توانایی دسترسی به سیستم را از هر جایی با استفاده از تکنولوژی‌های استاندارد شبکه‌ای (به عنوان مثال وب سایت‌ها) را چه از طریق کامپیوترهای رومیزی و چه از طریق دستگاه‌های موبایل داشته باشند.
- حفاظت از داده: داده یک نقش کلیدی و با ارزشی برای کاربران دارد بنابراین سیستم باید داده‌های را که مورد داده‌کاوی قرار می‌گیرند محافظت کند و دانش پی بردن به دسترسی غیرمجاز و تخریب‌های عمدی و تصادفی را داشته باشد.

۲-۲-۲-۲ نیازهای معماری

نیازهای غیرکاربردی اصلی در سطح معماری به سه دسته‌ی زیر تقسیم می‌شوند:

- جهت دهی سرویس: معماری باید به صورت مجموعه‌ای از اجزاء نرم‌افزاری تحت شبکه (سرویس‌ها) برای پیاده‌سازی قابلیت‌های عملیاتی مختلف سیستم طراحی شود تا استفاده‌ی مجدد، ترکیب و قابلیت همکاری آنها را به طور موثری قادر ساخت.
- باز و توسعه‌پذیر بودن: معماری باید برای ادغام با ابزار و سرویس‌های کشف دانش جدید باز باشد. علاوه بر این، سرویس‌های موجود طبق اصل باز و بسته بودن سرویس‌ها، باید برای توسعه باز ولی برای اصلاح بسته باشد.
- مستقل از زیرساخت بودن: معماری باید تا حد ممکن مستقل از زیرساخت طراحی شود؛ به عبارت دیگر، سرویس‌های سیستم باید قادر به بهره‌برداری از قابلیت‌های اساسی باشد که توسط زیرساخت‌های مختلف فراهم می‌شوند.

۳-۲-۲-۲ نیازهای زیرساخت

در نهایت، از دیدگاه زیرساخت، نیازهای غیرکاربردی زیر باید برآورده شود:

- دستیابی استاندارد شده: زیرساخت باید سرویس‌هایش را برای تکنولوژی‌های استاندارد (مانند سرویس‌های وب) باز بگذارد تا آنها را به عنوان بلوک‌های ساختمانی برای ایجاد سرویس‌های سطح بالا یا برنامه‌های کاربردی قابل استفاده کند.
- پشتیبانی از داده‌های توزیع شده و ناهمگون: زیرساخت باید قادر به پشتیبانی از پایگاه‌داده‌های خیلی بزرگ با ابعاد بزرگ باشد که با فرمت‌های مختلفی در یک مرکز داده ذخیره شده‌اند یا بر روی سایت‌های زیادی از نظر جغرافیایی توزیع شده‌اند.
- دسترس‌پذیری: زیرساخت باید در یک شرایط عملکرد باشد حتی اگر خرابی‌هایی رخ دهد که مجموعه‌ای از منابع سخت‌افزاری/نرم‌افزاری را تحت تاثیر قرار دهد. بدین ترتیب، روش‌های موثری (مانند افزونگی) باید پیاده‌سازی شود تا دسترسی قابل اعتمادی را برای منابع حساسی مانند داده‌های کاربر و برنامه‌های کاربردی تضمین کند.

- مقیاس‌پذیری: زیرساخت باید توانایی پشتیبانی از یک بارکاری¹ در حال رشد (ناشی از داده‌های بزرگ برای پردازش یا الگوریتم‌های سنگین برای اجرا) را به وسیله‌ی تخصیص پویای منابع مورد نیاز (پردازشگرها، منابع ذخیره‌سازی و شبکه‌ای) به طور موثر و کارایی داشته باشد. علاوه بر این، به محض اینکه بارکاری کاهش یافت، زیرساخت باید منابعی که لازم نیستند را آزاد کند.
- کارایی: زیرساخت باید مصرف منابع را برای اجرا شدن یک وظیفه به حداقل برساند. در مورد وظایف موازی/توزیع شده، تخصیص کارآمد گره‌های پردازشی باید تضمین شود. علاوه بر این، زیرساخت باید بیشترین بهره‌وری را داشته باشد تا سرویس‌های موثری را فراهم آورد.
- امنیت: زیرساخت باید روش‌های امنیتی موثری را فراهم آورد تا حفاظت داده‌ها، مدیریت هویت و حریم خصوصی را تضمین کند.

۲-۲-۳ مدل‌های ابری برای تحلیل داده‌های توزیع شده

همانطور که در بخش ۲-۱-۱ بحث شده است، فراهم‌کنندگان ابری سرویس‌های خود را به سه دسته‌ی اصلی طبقه‌بندی می‌کنند: نرم‌افزار به عنوان سرویس (SaaS)، که هر نرم‌افزار یا برنامه‌ی کاربردی اجرا شده از طریق اینترنت به مشتری‌ها به عنوان سرویس‌های آماده برای استفاده فراهم شده است. بستر به عنوان سرویس (PaaS)، فراهم نمودن بستر سرویس‌هایی از قبیل پایگاه‌داده‌ها، سرورهای برنامه‌های کاربردی، یا محیط‌های برای ساخت، آزمایش و اجرای برنامه‌های کاربردی مشتری؛ زیرساخت به عنوان سرویس (IaaS)، فراهم نمودن منابعی مانند CPUها، حافظه، و ذخیره‌سازی، برای اجرای سیستم‌های مجازی شده بر روی ابر.

سرویس‌های تحلیل داده برای برنامه‌های کاربردی KDD حساس به داده ممکن به صورت یکی از روشی که در زیر طبقه‌بندی شده‌اند پیاده‌سازی شوند:

- **KDD** به عنوان **SaaS**: در این مورد یک الگوریتم واضح و به خوبی تعریف شده‌ی داده‌کاوی یا یک ابزار کشف دانش آماده برای استفاده به عنوان یک سرویس اینترنتی فراهم شده است برای کاربران انتهایی که ممکن است مستقیماً از طریق یک مرورگر وب از این سرویس استفاده کنند.

¹ workload

- **KDD** به عنوان **PaaS**: در این مورد یک بستر پشتیبانی برای توسعه‌دهندگان فراهم شده است که قصد دارند برنامه‌های کاربردی خود را ایجاد یا برنامه‌های موجود را گسترش دهند.

توسعه‌دهندگان بدون نگرانی در مورد زیرساخت‌های اساسی یا مسائل محاسبات توزیع شده، تنها بر روی تعریف برنامه‌های کاربردی **KDD** خود تمرکز دارند.

- **KDD** به عنوان **IaaS**: در این مورد مجموعه‌ای از منابع مجازی به عنوان زیرساخت محاسباتی برای توسعه‌دهندگان فراهم شده است تا برنامه‌های کاربردی داده‌کاوی خود را اجرا کنند یا از ابتدا سیستم‌های **KDD** خود را پیاده‌سازی کنند.

در هر سه سناریوی بالا، ابر نقش فراهم‌کننده‌ی زیرساخت را بازی می‌کند حتی در موارد **SaaS** و **PaaS** لایه‌های زیرساخت می‌تواند برای کاربران انتهایی شفاف باشد.

به عنوان مثالی برای رویکرد **PaaS**، جدول ۲-۲ خلاصه‌ای را ارائه می‌دهد از اینکه چگونه اجزای **Microsoft Azure** و روش‌های آن که در بخش ۱-۳-۱-۲ معرفی شد، می‌توانند به طور موثری برای رفع نیازهای کاربردی یک سیستم تحلیل داده‌ی توزیع شده که در بخش ۱-۲-۲ مطرح شده‌اند، مورد استفاده قرار گیرند.

جدول ۲-۲: استفاده از **Microsoft Azure** برای رفع نیازهای کاربردی یک سیستم تحلیل داده‌ی توزیع شده

اجزای Microsoft Azure	نیازهای کاربردی
فرمت‌های مختلف داده‌ها: اشیاء بزرگ دودویی (Blobs)؛ جدول‌های غیررابطه‌ای (Tables)، صف‌هایی برای ارتباط داده‌ها (Queues)؛ پایگاه‌های داده‌های رابطه‌ای (SQL Database). پشتیبانی از فراداده ^۱ : جدول‌ها/ پایگاه داده‌های رابطه‌ای برای ذخیره‌ی تشریح داده‌ها؛ فیلدهایی که قابل تعریف توسط مشتری هستند می‌تواند به منابع داده‌ایی که شامل Blob ها هستند، اضافه شود.	داده‌ها مدیریت منابع
دسترسی مستقل از پیاده‌سازی: ابزاری که می‌توانند به عنوان وب سرویس ظاهر شوند. پشتیبانی از فراداده: جدول‌ها/ پایگاه داده‌های رابطه‌ای برای ذخیره‌ی	ابزار

¹ Metadata

<p>تشریح داده‌ها؛ فیلدهایی که قابل تعریف توسط مشتری هستند می‌تواند به منابع داده‌ایی که شامل Blobها هستند، اضافه شود؛ توصیفات WSDL برای وب سرویس‌ها.</p>		
<p>ذخیره‌سازی مدل‌ها: Blobها برای ذخیره‌ی نتایج هم به صورت متنی و هم به صورت تصویری. پشتیبانی از فراداده: جدول‌ها/ پایگاه داده‌های رابطه‌ای برای ذخیره‌ی تشریح داده‌ها؛ فیلدهایی که قابل تعریف توسط مشتری هستند می‌تواند به منابع داده‌ایی که شامل Blobها هستند، اضافه شود.</p>	نتایج	
<p>برنامه‌های کاربردی با تنها یک وظیفه: برنامه‌ریزی اجرای یک وب سرویس یا ابزار دودویی بر روی یک نمونه‌ی نقش کارگر. برنامه‌های کاربردی با پارامترهای فراگیر: برنامه‌ریزی اجرای همزمان مجموعه‌ای از وب سرویس‌ها یا ابزارهای دودویی بر روی مجموعه‌ای از نمونه‌های نقش کارگر. برنامه‌های کاربردی مبتنی بر گردش کار: برنامه‌ریزی اجرای هماهنگ مجموعه‌ای از وب سرویس‌ها یا ابزارهای دودویی بر روی مجموعه‌ای از نمونه‌های نقش کارگر.</p>	طراحی	مدیریت برنامه‌های کاربردی
<p>دسترسی منابع ذخیره‌سازی: توسط لایه‌ی ذخیره‌سازی مدیریت می‌شود. تخصیص منابع محاسباتی: توسط لایه‌ی محاسباتی مدیریت می‌شود. نظارت و اجرای برنامه‌های کاربردی: نمونه‌های نقش کارگر/ وب سرویس‌ها برای اجرای وظایف منفرد؛ جدول‌ها برای ذخیره کردن اطلاعات وظایف؛ نمونه‌ی نقش وب برای نمایش اطلاعات نظارتی. نمایش نتایج: جدول‌ها/ Blobها برای ذخیره کردن/تفسیر مدل‌های استنباطی؛ نمونه‌ی نقش وب برای نمایش نتایج.</p>	اجرا	

۲-۳ خلاصه

ابرها سرویس‌های پردازشی و ذخیره‌سازی مقیاس‌پذیری را فراهم می‌کنند که می‌توانند برای استخراج دانش از مخازن بزرگ داده و همچنین بسترهای نرم‌افزاری برای توسعه و اجرای محیط‌های تحلیل داده بر روی چنین سرویس‌هایی مورد استفاده قرار بگیرند. در این فصل مروری کلی بر روی تکنولوژی‌های ابری با استفاده از تشریح مدل‌های اصلی سرویس (نرم‌افزار به عنوان سرویس، بستر به عنوان سرویس، و زیرساخت به عنوان سرویس) و مدل‌های استقرار (ابرها، عمومی، خصوصی، یا آمیخته) که توسط فراهم‌کنندگان ابری اتخاذ شده‌اند، ارائه نمودیم. همچنین نمونه‌هایی از محیط‌های ابری (Microsoft Azure، Amazon Web Services، OpenNebula، و OpenStack) را شرح دادیم که می‌توانند برای پیاده‌سازی برنامه‌های کاربردی و چارچوب‌هایی برای تحلیل داده در ابر مورد استفاده قرار گیرند. در نهایت، پس از شناسایی نیازهای اصلی که باید توسط یک سیستم تحلیل داده‌ی توزیع شده برآورده شوند، به عنوان مثالی به شرح این مورد پرداختیم که چگونه اجزای Microsoft Azure و روش‌های آن می‌توانند برای برآورده نمودن این نیازها مورد استفاده قرار بگیرند.

مراجع

- R. Barga, D. Gannon, and D. Reed, "The client and the cloud: Democratizing research computing," *IEEE Internet Computing*, 15(1):72–75, 2011.
- Li, A., Yang, X., Kandula, S., Zhang, M., 2010. CloudCmp: comparing public cloud providers. Tenth ACM SIGCOMM Conference on Internet Measurement (IMC'10), New York, USA.
- Mell, P., Grance, T., 2011. The NIST Definition of Cloud Computing. NIST Special Publication 800-145.
- Nurmi, D., Wolski, R., Grzegorzczak, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D., 2009. The eucalyptus open-source cloud computing system. In: Proceedings of the Ninth IEEE/ ACM International Symposium on Cluster Computing and the Grid (CCGRID'09), Washington, USA.
- Richardson, L., Ruby, S., 2007. RESTful Web Services. O'Reilly & Associates, California.
- Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I., 2009. Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Comput.* 13, 14–22.