دوفصلنامه محاسبات و سامانه‌های توزیع شده
سال هفتم، شماره ۱، شماره پیاپی ۱۳، صفحه ۶۹ تا ۷۹، سال ۱۴۰۳

ارائه یک معماری جامع برای مرکز رصد خدمات ابری

داود ملکی (نویسنده مسئول)[۱]، پژمان گودرزی[۲]، سید محمدرضا میرصراف[۳]، یونس سیفی[۳]
[۱] عضو هیات علمی پژوهشگاه ارتباطات و فناوری اطلاعات تهران- ایران
[۲] دانشیار پژوهشگاه ارتباطات و فناوری اطلاعات تهران- ایران
[۳] استادیار پژوهشگاه ارتباطات و فناوری اطلاعات تهران- ایران
[۳] استادیار پژوهشگاه ارتباطات و فناوری اطلاعات تهران- ایران

چکیده

این مقاله با تشریح معماری سیستم نظارت بر ارائه‌دهنده خدمات ابری آغاز می‌شود و سپس توضیحاتی در مورد مرکز نظارتی که بر اساس آن معماری طراحی و اجرا شده است، ارائه می‌نماید. نظارت بر خدمات ابری برای اطمینان از سلامت ارائه‌های مبتنی بر ابر و همچنین برای حفظ عملکرد، امنیت و انطباق ضروری است. با اجرای یک مرکزپایش که شامل اجزاء، لایه‌ها و ابزارهای مختلف پیشنهاد شده در معماری است، سازمان‌ها می‌توانند مدیریت منابع ابری خود را بهبود دهند، سریع‌تر به حوادث واکنش نشان دهند و تعالی عملیاتی خود را حفظ یا بهبود بخشند. برای دستیابی به این هدف، لایه‌های نظارتی و ابزارهای سیستم‌های توزیع‌شده و APIهای مربوط به آن‌ها برای جمع‌آوری داده‌های مربوط به سیستم‌های رایانش ابری مورد بررسی و استفاده قرار گرفته‌اند. علاوه بر این، حوزه‌های خاصی از عملیات ابری برای نظارت بازبینی و تسهیل در نظر گرفته شده‌اند که در اجرای فرآیندهای مرتبط با سیستم نظارت ابری در مرکز رصد به کار گرفته می‌شوند.

کلمات کلیدی: رایانش ابری، پایش ابری، معماری مرکز رصد، مرکز رصد رایانش ابری

تاریخچه‌ی مقاله:
تاریخ ارسال: ۱۴۰۲/۱۲/۲۵
تاریخ اصلاحات: ۱۴۰۳/۰۵/۲۰
تاریخ پذیرش: ۱۴۰۳/۰۶/۲۸
تاریخ انتشار: ۱۴۰۳/۰۶/۳۰
ایمیل نویسنده مسئول:dmaleki@itrc.ac.ir

۱ - مقدمه

رصد ابری عبارت از عمل اندازه‌گیری، ارزیابی، نظارت و مدیریت بارهای کاری در محیط‌های چند کاربره ابری با بکارگیری معیارها و آستانه‌های مشخص می‌باشد. رصد می‌تواند بصورت دستی یا خودکار و به منظور تأییدِ در دسترس بودن کامل ابر و درستی کارکرد ابر، استفاده گردد. پایش ابری امکان بررسی انطباق عملکرد برنامه‌های میزبانی‌شده در فضای ابری با توافق‌نامه سطح سرویس (SLA) را فراهم می نماید. ضمناً به کمک آن می‌توان خطرات امنیتی احتمالی را کشف، مشکلات ظرفیت را شناسایی و نهایتا هزینه‌ها را تجزیه‌وتحلیل نمود. در راستای دستیابی به حالت ایده‌آل، رصد و پایش ابری در کنار سیستم‌های پایش داخلی رایانش ابری و بصورت زمان واقعی کار می‌کنند. این امر به بهبود نظارت بر تمامی اجزاء از جمله فضای ذخیره‌سازی، شبکه‌ها و برنامه‌ها کمک می‌نماید. ضمناً می‌توان قابلیت‌های دیگر ابزارهای رصد ابری، همچون ردیابی مصرف و ترافیک منابع میزبان ابری را نیز بکار گرفت. قابلیت اندازه‌گیری و تجسم عملکرد لایه شبکه بین ابر ترکیبی، ابر خصوصی و سرویس‌های ابری عمومی برای رصد ابری بسیار اهمیت دارد. از دیگر کاربردهای ابزار رصد می‌توان به مواردی مانند یکسانسازی حجم زیادی از داده‌ها در مکان‌های توزیع‌شده، شناسایی ناهنجاری‌ها و علل ریشه‌ای آن‌ها، پیش‌بینی خطرات بالقوه، قطع ارائه خدمات، قابلیت اندازه‌گیری و بصری‌سازی عملکرد لایه شبکه بین ابر ترکیبی، ابر خصوصی و سرویس‌های ابری عمومی اشاره نمود.

مرکز رصد رایانش ابری دارای اجزاء و ابزارهای متعددی می‌باشد. در این مراکز انواع ابزارهای خودکار حوزه‌های مختلف عملکرد را ردیابی می‌کنند. برخی از ابزارها بصورت خاص منظوره برای خدمات ابری ساخته شده‌اند. برخی دیگر در سیستم‌عامل‌های توسعه داده شده توسط طرف‌های ثالث عرضه می‌شوند. در تمامی این موارد، بهترین راه‌حل رصد ابری باید شامل معیارهای سفارشی باشد که می‌تواند بخش‌های خاصی از پشته ابر و همچنین محیط را به‌طورکلی نظارت کند [۱]. با توجه به مطلب بیان شده می‌توان دریافت که راهکار «رصد» به یک بخش مهم و حیاتی در رایانش ابری تبدیل‌شده است [۲]. مزیت اصلی آن، شناسایی زودهنگام خرابی و اطلاع‌رسانی به تیم‌های پشتیبانی از طریق ایمیل یا پیام متنی است. برخی از این راه‌حل‌ها و راهکارها عملکردی خودکار را برای رفع فوری مشکل ارائه می‌کنند؛ ازاین‌رو خرابی سیستم و دخالت انسان را به حداقل می‌رسانند. نمونه‌هایی از راهکارهای رصد Zabbix و Ganglia می‌باشند. Zabbix سیستمی را کنترل و رصد می‌کند که در سازمان

اروپایی تحقیقات هسته‌ای (CERN) [۳]. استفاده می‌شود. ضمناً Ganglia دایرۀالمعارف اینترنتی محتوای آزاد ویکی‌پدیا [۴] را رصد می‌کند.

۲ – کارهای مرتبط

نظارت بر زیرساخت‌های رایانش ابری برای ارائه‌دهنده‌ها و مدیران ابری برای تحلیل، بهینه‌سازی و کشف آنچه در زیرساخت‌ها اتفاق می‌افتد، یک ضرورت اساسی است. ساختار معماری رصدی نظارت بر ماشین‌های فیزیکی و مجازی موجود در یک زیرساخت ابری را به روشی غیرتهاجمی، شفاف فراهم می‌کند و آن را نه‌تنها برای محاسبات ابری خصوصی، بلکه برای زیرساخت‌های محاسبات ابری عمومی نیز فراهم می‌کند. معماری Nagios با استفاده از یک نمونه اولیه که یک راه‌حل رصدی موجود در کلاس سازمانی با یک پشته شناسایی می‌باشد و همچنین OpenStack برای طراحی صفحات کنترل و داده برای زیرساخت‌های ابری استفاده می‌شود [۵]. درنتیجه، با استفاده از معماری Nagios می‌توان عملکردهای خروجی را گسترش داد تا در نظارت بر زیرساخت‌های ابری نقش ایفا کند معماری پیشنهادی در مقاله [۵] به‌صورت متن‌باز طراحی، پیاده‌سازی شده است. نرم‌افزار به‌صورت مداوم نرم‌افزار -منابع سرور- شبکه و کارهایی مثل میزان استفاده از حافظه -بار ریزپردازنده- تعداد پردازنده‌ها را چک می‌کند. هم‌چنین می‌تواند سایر سرویس‌ها مانند پروتکل انتقال ایمیل، پروتکل‌های HTTP و سایر پروتکل‌های استاندارد موجود شبکه را بررسی کند. بررسی‌های فعال و مهم توسط Nagios انجام می‌شود، درحالی‌که سایر چک‌های ثانویه توسط برنامه‌های ثانویه مرتبط با ابزار پایش انجام می‌شود.

این نرم‌افزار می‌تواند کنترل‌کننده رویداد را که در زمان وقایع میزبان یا خدمات برای حل مشکلات پیشگیرانه اجرا می‌شود توصیف کند. هم‌چنین برای پشتیبانی از افزونگی در نظارت بر میزبان استفاده می‌شود.

هم‌چنین می‌تولند در ابزارهای سخت‌افزاری مانند کاوشگر برای هشدار، دمایی که می‌تواند اطلاعات جمع‌آوری‌شده را از طریق شبکه با پلاگین‌های پیکربندی‌شده، کنترل کند. رصد از راه دور می‌تواند از طریق مجری پلاگین راه دور Nagios از طریق کلنال‌های رمزگذاری شده SSL و SSH ایجاد شود [۶].

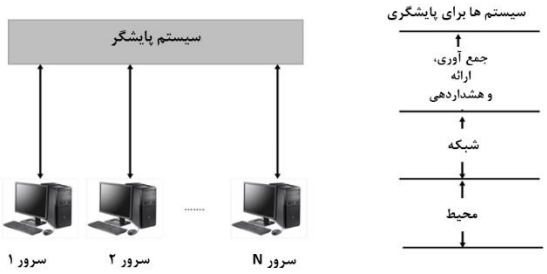
این فرآیند را بر روی backend پایگاه داده انجام می‌دهد، داده‌ها را ترسیم می‌کند و افزونگی را در زمان نظارت بر میزبان اعمال می‌کند. رابط وب برای مشاهده وضعیت فعلی شبکه، تاریخچه مشکل، مدیر اعلان، پرونده ها، گزارش‌ها و غیره است.

معماری میکروسرویس، برخلاف معماری یکپارچه مانند Nagios، که برنامه را به‌عنوان یک «کلِ واحد» در نظر می‌گیرد، برنامه را به بخش‌های مختلف و کوچک‌تری تقسیم می‌کند. در این نوع معماری هر پروسه از برنامه به‌عنوان یک سرویس جداگانه در نظر گرفته می‌شود بنابراین هرکدام از این سرویس‌ها منطق و پایگاه داده خودشان را دارند و کار خاص خودشان را انجام می‌دهند. باید در نظر داشت که در این تعریف وقتی از «بخش‌های کوچک‌تر» صحبت می‌شود، منظور ماژول‌هایی مستقل است که به‌صورت جداگانه deploy می‌شوند و به یکدیگر هیچ نیازی ندارند اما برای تشکیل یک برنامه بزرگ‌تر از طریق API با یکدیگر تعاملات خودشان را دارند [۷].

برخی از مزایای معماری میکروسرویس، درک ساده تر، مقیاس پذیری و انعطاف در انتخاب فناوری می‌باشد. در این مقاله، معماری ارائه شده براساس مزایای معماری میکروسرویس، طراحی و ارائه شده است.

۳ - لایه‌های پایشگری

لایه‌های پایشگری سامانه رصد ابری در شکل ۱ نمایش داده شده است. این لایه‌ها شامل سه لایه زیرساخت (محیط) ، شبکه و جمع‌آوری، ارائه و هشداردهی می‌باشند.



شکل ۱: لایه‌های پایشگری در سامانه رصد ابری [۱]

- **زیرساخت (محیط):** این لایه، زیرساخت سیستم‌های توزیع‌شده را نشان می‌دهد و شامل انواع مختلفی از سخت‌افزار، سیستم‌عامل، برنامه‌های کاربردی و خدمات است. بعلاوه دستگاه‌های ذخیره‌سازی و پایگاه‌های داده را نیز شامل می‌شود.
- **شبکه:** این بخش از نمودار جریان داده وظیفه انتقال داده‌های رصدی از یک منبع (سیستم رصدشده) به یک مقصد (سیستم رصد) را بر عهده دارد. شبکه می‌تواند به‌صورت محلی (شبکه محلی) یا دارای توزیع جغرافیایی (شبکه گسترده) نگریسته شود .
- **جمع‌آوری، ارائه و هشداردهی:** لایه اصلی که رصد را آغاز می‌کند، داده‌های جمع‌آوری‌شده را ذخیره می‌کند، معیارها را بصری نموده و در صورت نیاز اعلان‌های هشدار را راه‌اندازی و یا صادر می‌کند. در این قسمت کاربر نهایی کل راه‌حل را پیکربندی و کنترل می‌کند و هم‌چنین گزارش‌های مرتبط با معیارهای در دسترس بودن و ظرفیت را تهیه و اجرایی می‌کند.

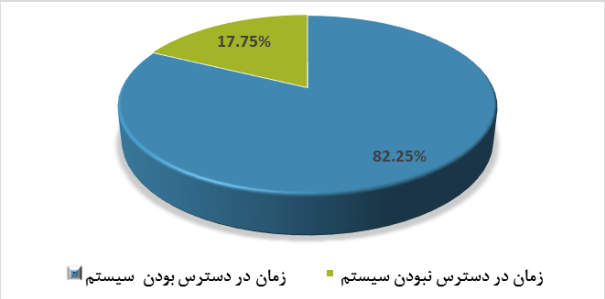
۳-۱: حوزه‌های رصد

در زیرساخت دستگاه‌های مرتبط با رایانش توزیع‌شده و ابری، حوزه‌های متعددی وجود دارند که باید معیارهای آن‌ها جمع‌آوری شده و مورد رصد و پایش قرار گیرند. بخش اصلی راه‌حل‌های ارائه شده به منظور «رصد در دسترس بودن»، «رصد ظرفیت و عملکرد»، و «رصد رویدادهای امنیتی» طراحی‌شده است.

- **رصد در دسترس بودن:**

حوزه «رصد در دسترس بودن» جزئیاتی را در مورد سیستم‌ها، خدمات و میزان دسترسی به برنامه ارائه می‌دهد و بر اساس درصد مدت زمانی که سیستم در دسترس است و درصد مدت زمانی که سیستم در دسترس نیست تعیین می‌گردد. شکل ۲، نمونه‌ای از رصد خدمت در دسترس بودن در بازه زمانی عملکرد سیستم که ۲۴ ساعت از روز و ۷ روز از هفته می‌باشد را نمایش می‌دهد.

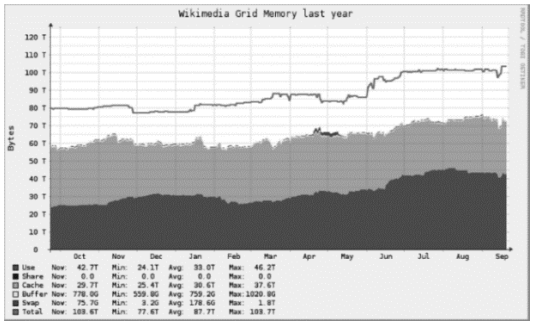
امروزه دستگاه‌های متعددی طراحی‌شده‌اند که به‌صورت ۲۴ ساعته و در تمامی ۷ روز هفته کار می‌کنند بااین‌حال دستیابی به دسترسی ۱۰۰ درصد، در مدت زمانی یکسال بسیار دشوارمی‌باشد.



شکل ۱: رصد خدمات در دسترس بودن طی زمان عملکرد سیستم ۲۴ ساعت در ۷ روز هفته

- **رصد ظرفیت و عملکرد**

دومین حوزه‌ای که مورد رصد قرار می‌گیرد «رصد ظرفیت و عملکرد» است. با رشد روزافزون دستگاه‌های توزیع‌شده و هم‌چنین بکارگیری منابع کامپیوتری بیشتر، این حوزه و معیارهای مرتبط با آن به پیش‌بینی نیازهای آینده در مورد توان محاسباتی کمک می‌کند. به‌طورکلی این سنجه شامل CPU، حافظه و منبع ذخیره‌سازی است. شکل ۳ ظرفیت نمونه بر اساس حافظه شبکه ویکی‌پدیا در ابزار Ganglia را نشان می‌دهد.



شکل ۲: نمونه رصد ظرفیت در ابزار Ganglia [۴]

- **رصد رویدادهای امنیتی**

حوزه «رصد رویدادهای امنیتی»، راه‌حل‌های اختصاصی برگرفته از اطلاعات امنیتی و مدیریت رویداد (SIEM[1]) را ارائه می‌دهد. در این حوزه همه زیرساخت‌ها و ابزارهای رصد، دارای ماژول سفارشی مرتبط با خود هستند که به کمک آن می‌توانند تأیید رویدادهای امنیتی اولیه را ارائه دهند [۸] و در راستای هدف اصلی رصد که جمع‌آوری رویدادها از لاگ‌های امنیتی و فایروال‌ها، ذخیره و نهایتاً تجزیه‌وتحلیل آن‌ها می‌باشد مورد استفاده قرار گیرند. مثال‌هایی از رویدادهای امنیتی تلاش برای دسترسی غیرمجاز از طریق وارد نمودن رمز عبور اشتباه یا عدم دسترسی به مجوزها و یا انکار سرویس توزیع‌شده (DDoS) است. علاوه بر آن راهکار پایش خدمات ابری می‌تواند تعداد افزایش تلاش به منظور ورود به سیستم از یک مکان خاص را شناسایی نموده و قوانین فایروال مربوطه را آنگونه تنظیم نماید که ترافیک متفرقه شبکه مسدود گردد.

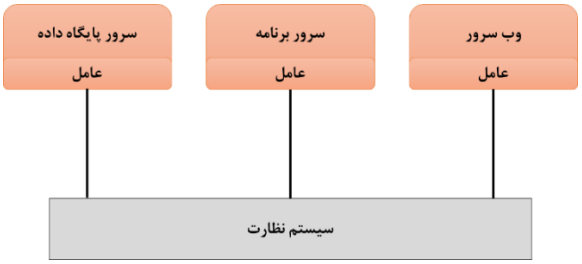
۴ – رویکردهای رصدی

رویکردهای رصدی رایج دو نوع، مبتنی بر عامل و بدون عامل می‌باشند. البته اخیراً روش‌های جدیدی معرفی‌شده‌اند که مزایای مبتنی بر عامل و غیر مبتنی بر عامل را در یک رویکرد ترکیبی در کنار هم قرار داده یا مجموعه‌ای از معیارهای رصد از طریق جریان‌های داده را در برمی‌گیرد [۸].

۴-۱: رویکرد مبتنی بر عامل

یک «رویکرد مبتنی بر عامل» وابسته به پلتفرم است و به نرم‌افزار اضافی در سیستم‌های رصدشده نیاز دارد که در آن عامل برنامه کاربردی، پایگاه داده، و مقیاس‌پذیری سیستم‌ها به‌صورت پلتفرم طراحی‌شده‌اند. شکل ۴ معماری رویکرد مبتنی بر عامل در یک سیستم رصدشده را نشان می‌دهد.

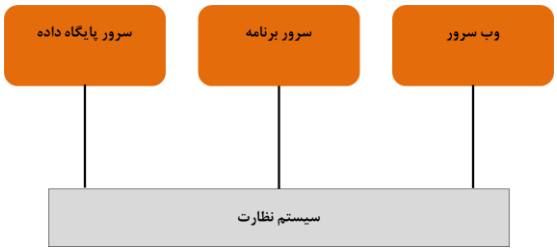
[1] Security information and event management



شکل ۳: معماری رویکرد مبتنی بر عامل سیستم رصدشده [۱]

۴-۲: رویکرد غیر مبتنی بر عامل

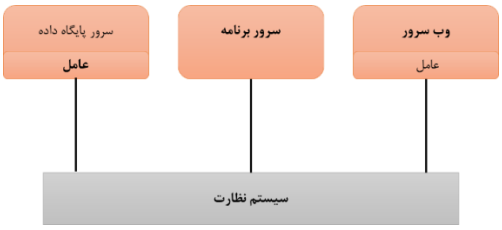
یک «رویکرد غیر مبتنی بر عامل» از فناوری‌ها و پروتکل‌های رصد داخلی سیستم‌هایی مانند ابزار مدیریت ویندوز و پروتکل مدیریت شبکه ساده (SNMP) در دسترس، استفاده می‌کند. این راه حل، از آنجایی‌که برای نصب، نیاز به نرم‌افزار اضافی ندارد و هم‌چنین استقرار آن در محیط توزیع‌شده ساده‌تر هست روشی آسان می‌باشد. همان‌طور که در شکل ۵ نشان داده شده، رویکرد غیر مبتنی بر عامل، رصد میزان در دسترس بودن سیستم‌ها را بدون ماژول‌های اضافی، مانند آنچه در رویکرد مبتنی بر عامل (شکل ۴) مورد نیاز است، فراهم می‌کند. با این‌حال یک رویکرد غیر مبتنی بر عامل به معیارهای رصد عمومی محدود می‌شود.



شکل ۴: معماری رویکرد غیر مبتنی بر عامل [۱]

۴-۳: رویکرد ترکیبی

«رویکرد ترکیبی»، با ترکیب نمودن روش‌های مبتنی بر «رویکرد عامل» و «رویکرد غیر مبتنی بر عامل» طراحی می‌شود. در رویکرد ترکیبی، بخش رصد مبتنی بر عامل را می‌توان بر روی دستگاه‌های حیاتی نصب کرد. در این دستگاه‌ها داده‌های رصد در فواصل زمانی کوتاه و به منظور به حداقل رساندن زمان توقف برنامه‌ها مورد نیاز است. بخش رویکرد غیر مبتنی بر عامل را نیز می‌توان بر روی دستگاه‌های استاندارد استفاده نمود. دستگاه‌های استاندارد کاربردی دارند که در آن تنها معیارهای اساسی، مانند در دسترس بودن سیستم و CPU و استفاده از دیسک مورد نیاز است.



شکل ۵:: معماری رویکرد ترکیبی [۱]

رویکرد ترکیبی با بکارگیری تکنیک سنتی مربوطه اجازه می‌دهد تا انطباق با نیازهای کسب‌وکار بهتر صورت پذیرد و این انطباق با رشد زیرساخت حفظ شود. شکل ۶ معماری رویکرد ترکیبی را نشان می‌دهد.

۴-۴: ابزارهای رصدی سیستم‌های توزیع شده

ابزارهای تجاری و غیرتجاری متعددی در بازار وجود دارند که دستگاه‌های توزیع‌شده را با استفاده از رویکرد مبتنی بر عامل، بدون عامل، رویکرد ترکیبی و رویکرد جریان داده رصد می‌کنند. جدول ۱ به بررسی ۱۵ مورد از این ابزارهای رصدی می¬پردازد [۱].

جدول ۱: بررسی ۱۵ ابزار نظارتی برای سیستم‌های توزیع شده

ابزار	بازار هدف (اندازه)	ابر سفارشی	مکانیزم هشدار	رویکرد رصد	مجوز	ویژگی‌ها
Zabbix	شرکت و بزرگ	بله	سفارشی,Email,SMS	Agent-based, agentless	Open source / اختصاصی	کشف خودکار پلاگین‌های چندگانه
Nagios	شرکت متوسط کوچک و بزرگ	بله	سفارشی,Email,SMS	Agent-based, agentless	Open source / اختصاصی	افزونه‌های چندگانه، جامعه پشتیبانی گسترده، جامعه نسخه‌های سفارشی‌شده
Ganglia	شرکت و بزرگ	بله	Nagios اختیاری از طریق	Agent-based	Open source	خوشه‌ها و پشتیبانی از شبکه
Hyperic	کوچک و متوسط	بله	Email, SMS	Agent-based, agentless	Open source / اختصاصی	استقرار آسان و پیکربندی
ManageEngine	کوچک و متوسط	بله	سفارشی,Email,SMS	Agentless	اختصاصی	استقرار سریع و آسان، کاربرد نظارت عملکرد
AppManager	تصدی	بله	Email,SMS	Agent-based, agentless	اختصاصی	پیش‌بینی‌کننده، تجزیه‌وتحلیل و تهیه گزارش
نظارتIBM	تصدی	نه	سفارشی,Email,SMS	Agent-based, agentless	اختصاصی	یکپارچه‌سازی با سایر محصولاتHP
SmartCloud	شرکت و بزرگ	بله	Email,SMS,API	Data streams	اختصاصی	نرم‌افزار به‌عنوان یک خدمت
HP Operation Manager	شرکت متوسط کوچک و بزرگ	بله	Email,SMS,API	Data streams	اختصاصی	پشتیبانی از سکوهای ابری چندگانه و همکاری تیم‌هایDev Ops

AppDynamics	شرکت و بزرگ	بله	NO	Data streams	Open source	می‌تواند در هر دقیقه ۱۶۰ هزار معیار متمایز را اداره کند
Datadog	شرکت متوسط کوچک و بزرگ	بله	Email,SMS,API	Data streams	اختصاصی	رصد مصنوعی، نرم‌افزار به‌عنوان خدمات
Graphite	شرکت متوسط کوچک و بزرگ	بله	Email,SMS,API	Data streams	Open source	فعال جامعه از توسعه‌دهندگان و کاربران
New Relic	کوچک و متوسط	بله	Email,SMS,API	Data streams	Open source	پشتیبانی از جامعه گسترده
Prometheus	شرکت متوسط کوچک و بزرگ	بله	Email,SMS,API		Open source / اختصاصی	پیکربندی و اتوماسیون فایل‌ها
Riemann	شرکت متوسط کوچک و بزرگ	بله	Email,SMS,API	Data streams	Open source / اختصاصی	نرم‌افزار به‌عنوان خدمت

۴-۵: حوزه‌های عملیات ابری تسهیل شده با نظارت

در حالی که مصرف‌کنندگان ابری از نگرانی‌های مربوط به هزینه‌های نگهداری آزاد هستند، ارائه‌دهندگان از طرف دیگر مسئول نگهداری و مدیریت زیرساخت‌های اصلی هستند. نظارت بخش ضروری مدیریت ابری بوده و اهداف مختلفی مانند (۱) تأمین منابع/خدمات، (۲) برنامه‌ریزی بهینه ظرفیت، (۳) اطمینان از توافق‌نامه‌های سطح خدمات (SLA)، (۴) مدیریت پیکربندی، (۵) صورتحساب و (۶) اطمینان از امنیت/حریم خصوصی را برای ارائه‌دهندگان ابری شامل می‌شود.

مطلب فوق مناطق عملیاتی ابری را که با نظارت تسهیل می‌شوند، مورد تأکید قرار می‌دهد. جدول ۲ این مناطق را خلاصه کرده و نشان می‌دهد که کجا ارائه‌دهندگان و مصرف‌کنندگان دیدگاه‌های متفاوتی در مورد نظارت دارند. در ادامه، مناطق عملیاتی ابری که می‌توانند از نظارت بهره‌مند شوند توصیف می گردند. در این فرآیند، به قابلیت‌های مطلوب ابزار نظارتی که آن را برای هدف مورد نظر مناسب می‌کند، پرداخته می‌شود.

جدول ۲:مناطق عملیات ابری از دید ارائه‌دهنده و مصرف کننده

حوزه‌های عملیات ابری: دیدگاه‌های فراهم آورنده و مصرف کننده		
حوزه‌های عملیات ابری	دیدگاه فراهم آورنده	دیدگاه مصرف کننده
حسابداری و صورتحساب	بله	بله
مدیریت SLA	بله	بله
تامین خدمات/ منابع	بله	خیر
برنامه‌ریزی ظرفیت	بله	خیر
مدیریت پیکربندی	بله	خیر
اطمینان از امنیت و حریم خصوصی	بله	بله
مدیریت خطا	بله	خیر

- **حسابداری و صورتحساب**

مفهوم ارائه خدمات محاسباتی به عنوان یک سرویس عمومی، به شدت به توانایی ثبت و گزارش اطلاعات استفاده از ابر که براساس آن طرح‌های صورتحساب پایه‌ریزی می‌شود، وابسته است. حسابداری و صورتحساب دقیق به توانایی ضبط اطلاعات مصرف و تخصیص منابع مجازی و همچنین اطلاعات برنامه‌ها (به عنوان مثال، ساعات محاسباتی استفاده شده، پهنای باند استفاده شده) نیاز دارد [۹]. علاوه بر این، ارائه یک سیستم صورتحساب شفاف که قادر به ثبت داده‌ها به صورت قابل تأیید و مطمئن باشد، برای اطمینان از محافظت در برابر جعل و تغییرات کاذب، نیازمند قابلیت‌های نظارتی قوی و ایمن است [۱۰،۱۱].

- **مدیریت SLA**

یک توافقنامه سطح خدمات (SLA) قراردادی است که بین ارائه‌دهنده خدمت و مشتری امضا می‌شود و شرایط ارائه خدمت از جمله کیفیت سرویس (QoS)، قیمت‌گذاری و جریمه‌ها در صورت نقض شرایط توافق شده را مشخص می‌کند [۱۲-۱۳]. مدیریت SLA برای ارائه‌دهندگان ابر بسیار حائز اهمیت است زیرا اطمینان از اجرای SLA برای جلب رضایت مشتری الزامی است. قابلیت‌های نظارت لازم برای پشتیبانی از عملیات در این زمینه شامل توانایی اندازه‌گیری پارامترهای QoS، ذخیره و تحلیل داده‌ها، اندازه‌گیری مصرف منابع و ارزیابی پارامترهای SLA است [۱۲-۱۵].

- **تأمین خدمات/منابع**

تأمین خدمات/منابع شامل تخصیص بهینه منابع به منظور هماهنگی با بار کاری است [۱۶]. این امر نیازی اساسی برای فراهم

کردن انعطاف‌پذیری ابری است. تأمین منابع به دو روش پیاده‌سازی می‌شود: (۱) تأمین استاتیک که در آن ماشین‌های مجازی (VM) با اندازه مشخصی ایجاد می‌شوند و سپس بر روی مجموعه‌ای از سرورهای فیزیکی تجمیع می‌شوند. و ظرفیت VM تغییر نمی‌کند؛ و (۲) تأمین پویا که ظرفیت VM به طور پویا تنظیم می‌شود تا با نوسانات بار کاری تطابق پیدا کند [۱۷]. توانایی اندازه‌گیری مصرف کلی منابع یک سیستم، همراه با توانایی اندازه‌گیری مصرف منابع هر سرویس (که مقدار منابع مورد نیاز هر سرویس مجزا را شناسایی می‌کند)، به منظور تأمین موثر، ضروری است. بعلاوه، توانایی ارزیابی ریسک و کیفیت خدمات (QoS) برای اتخاذ تصمیم‌گیری‌های موثر در تأمین منابع، مانند اینکه آیا منابع را تخصیص دهیم یا آزاد کنیم تا کیفیت تحت تاثیر قرار نگیرد یا منابع هدر نرود، مورد نیاز است [۱۸،۱۶].

- **برنامه‌ریزی ظرفیت**

برنامه‌ریزی ظرفیت، به ویژه برای ارائه‌دهنده یک حوزه مهم در رایانش ابری است. از این طریق اطمینان حاصل می‌گردد که منابع کافی برای برآورده کردن تقاضای ظرفیت که برای تأمین سطح کیفیت خدمات و انجام فعالیت‌های مختلف مدیریت عملیاتی ابری، مانند بازیابی از حوادث و نگهداری از پشتیبان‌ها لازم است، وجود دارد [۱۹]. وجود امکان اندازه‌گیری میزان استفاده از ظرفیت این امکان را فراهم می‌کند که اقداماتی مانند پیش‌بینی نیاز به منابع بیشتر یا تعیین میزان هدر رفت منابع، قابل انجام باشند. بعلاوه، توانایی تشخیص در دسترس بودن گره‌های ابری برای حفظ سطحی مورد نیاز از محدودیت‌های منابع لازم است [۲۰].

- **مدیریت پیکربندی**

پیکربندی مجموعه‌ای از پارامترها و مقادیر است که رفتار دستگاه‌ها و نرم‌افزارها را تعیین می‌کند [۲۱]. از آنجایی که ارائه‌دهنده‌ی ابری ممکن است در محیطی چندمستأجره فعالیت کند، نیاز دارد که پیکربندی‌های خاص مشتری را مدیریت کند. سیستم مدیریت پیکربندی باید قادر باشد پیکربندی‌های مشخص شده را تأیید کرده و تغییرات ممکن را شناسایی کند [۲۱-۲۲]. در این زمینه، سیستم نظارت از پشتیبانی‌هایی که در ادامه بیان می‌شوند برخوردار است: تأیید پیکربندی با شناسایی یک پیکربندی و تأیید اینکه نمونه‌ها دارای پیکربندی مورد نظر هستند؛ شناسایی انحراف پیکربندی با تعیین اینکه آیا پیکربندی یک نمونه تغییر کرده است؛ و نظارت بر تأثیر پیکربندی با حسابرسی اثرات عملکرد ناشی از تغییرات پیکربندی.

- **اطمینان از امنیت و حریم خصوصی**

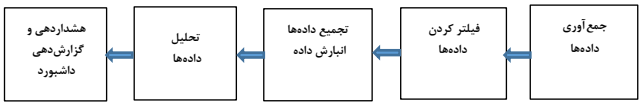
ریسک از دست دادن امنیت و حریم خصوصی یکی از موانع اصلی در برابر استفاده گسترده از محاسبات ابری است. برای اطمینان از امنیت خدمات ابری، مهم است که اطمینان حاصل شود که ابزاری که برای نظارت استفاده می‌شود هیچ آسیب‌پذیری‌ای به وجود نیاورد. این موضوع به‌ویژه در یک محیط ابری چندمستاجری مهم است.

- **مدیریت خطا**

مدیریت خطا یکی از حوزه‌های اصلی عملیاتی در ابر است که خدمات ابر قابل اعتماد و مقاوم را فراهم می‌کند [۲۳]. نظارت مداوم امکان پیش‌بینی و شناسایی به‌موقع خرابی‌ها را فراهم می‌کند، که می‌توان آن‌ها را به‌صورت پیشگیرانه با تعویض اجزای مشکوک مدیریت کرد [۲۴]. به‌دلیل ترکیب پیچیده اجزای ابر، خطاها می‌توانند به روش‌های مختلفی مانند بار زیاد روی سرور یا خرابی شبکه/سخت‌افزار/خدمات رخ دهند [۲۵]. بنابراین، قابلیت شناسایی بار یک سیستم ابر و تشخیص در دسترس بودن اجزا برای حمایت از عملیات در این حوزه ضروری است.

۵ – ساختار معماری پیشنهادی

فرایند نظارت در سیستم نظارت ابری (CMS[2]) را می‌توان به پنج مرحله متمایز تقسیم کرد: ۱) جمع‌آوری داده‌ها، ۲) فیلتر کردن داده‌ها، ۳) تجمیع داده‌ها، ۴) تحلیل داده‌ها و ۵) هشداردهی و گزارش‌دهی. شکل ۷ این مراحل فعالیت نظارت ابری را نشان می‌دهد. این بخش تعریفی مختصر از هر مرحله ارائه می‌دهد.



شکل ۶: مراحل فعالیت نظارت ابری

۵-۱: جمع‌آوری داده‌ها

سیستم نظارت باید انواع مختلف اطلاعات یا معیارهایی مانند زمان پردازش، سرعت پردازش CPU، استفاده از حافظه، تأخیر حافظه، مصرف انرژی، استفاده از انرژی، پهنای باند، تأخیر و غیره را از پروب‌های مختلف جمع‌آوری کند.

- **معماری**

[2] Cloud Monitoring System

داده‌ها می‌توانند با استفاده از معماری متمرکز یا غیرمتمرکز جمع‌آوری شوند. معماری متمرکز از درخت‌های جمع‌آوری داده استفاده می‌کند که در آن یک سرور رصد در ریشه درخت قرار دارد و از سرورهای سطح پایین‌تری که وضعیت را از میزبان‌های پایش منتقل می‌کنند، پشتیبانی می‌شود. خرابی یک سرور پایش، فرآیند جمع‌آوری داده‌ها از زیربنای آنرا متوقف خواهد کرد. سیستم جمع‌آوری داده متمرکز با چالش‌های مختلفی همچون نقطه واحد خرابی، کاهش عملکرد، تکرار و تحمل خطا مواجه است. برای غلبه بر این چالش‌ها، سیستم پایش باید به یک سیستم غیرمتمرکز روی آورد. سیستم غیرمتمرکز از مفهوم همتا به همتا استفاده می‌کند. این سیستم از جمع‌آوری داده مبتنی بر عامل یا جمع‌آوری داده بدون عامل بهره می‌برد. در جمع‌آوری داده مبتنی بر عامل، عامل‌ها در اجزای مختلف ابر نصب می‌شوند تا داده‌ها را جمع‌آوری و به سرور مرکزی ارسال کنند. در CMS بدون عامل، جمع‌آوری داده ساده‌تر و ارزان‌تر است، زیرا نصب نرم‌افزار عامل ضروری نیست.

- **استراتژی**

برای جمع‌آوری داده‌ها از ابر، پنج استراتژی شناسایی شده است: ۱) فشار ۲) کشش ۳) فشار ترکیبی، ۴) کشش-فشار ترکیبی و ۵) فشار تطبیقی. روش فشار، اطلاعات را از اجزای انتهایی ابر به سرور رصد منتقل می‌کند. در روش کشش، سرور رصد از اجزای انتهایی ابر درخواست می‌کند تا اطلاعات مورد نیاز را ارسال کنند. روش فشار ترکیبی داده‌ها یا اطلاعات را به گره رصدی بر اساس بازه‌های زمانی ثابت یا بر اساس رویدادی که در ابر رخ می‌دهد، منتقل می‌کند. در روش کشش-فشار ترکیبی، یک نهاد داده‌ها را از نهادهای خارجی می‌کشد و همان داده‌ها را به گره رصدی منتقل می‌کند. فشار تطبیقی داده‌ها را از پروب‌های مختلف جمع‌آوری کرده و در یک پنجره واحد ذخیره می‌کند. در معماری پیشنهادی، برای جمع‌آوری داده مبتنی بر این استراتژی‌ها از میکرو سرویس‌های معماری استفاده می‌شود که با توجه به وضعیت سنجه مورد اندازه گیری می‌تواند با فراخوانی رابط برنامه‌نویسی اپلیکیشن (API) با اجزای انتهایی ابر ارتباط بر قرار کرده و در فرایندهای مبتنی بر عامل و غیر مبتنی بر عامل و همچنین فرایند های ترکیبی به جمع‌آوری سنجه‌ها بپردازد.

- **تکنیک‌های به‌روزرسانی**

اطلاعات در ابر به‌طور مکرر به‌روزرسانی می‌شود. بنابراین، ضروری است که داده‌های به‌روز برای تحلیل در نظر گرفته شوند. چهار تکنیک به‌روز رسانی در کارهای موجود شناسایی شده است. اول، داده‌هایی که در فواصل منظم به‌روزرسانی می‌شوند، و به‌عنوان به‌روزرسانی دوره‌ای نامیده می‌شوند. دوم، داده‌ها زمانی جمع‌آوری می‌شوند که رویدادی در ابر به وقوع بپیوندد که به‌عنوان به‌روزرسانی مبتنی بر رویداد شناخته می‌شود. سوم، داده‌های جدید تنها زمانی به‌روزرسانی می‌شوند که تغییراتی در داده‌های جمع‌آوری شده قبلی وجود داشته باشد که به‌عنوان به‌روزرسانی مبتنی بر محتوا شناخته می‌شود. چهارم، سیستم مبتنی بر پنجره، داده‌ها را تا زمان ورود داده‌های جدید در پنجره ذخیره می‌کند و سپس داده‌های پنجره را با داده‌های جدید مقایسه می‌کند و فقط در صورت وجود تغییر در داده‌ها، اطلاعات به میکرو سرویس‌های مرکز رصد ارسال می‌شود.

۵-۲: فیلتر کردن داده‌ها

اطلاعات جمع‌آوری‌شده ممکن است حاوی درصد بالایی از داده‌های تکراری، نامعتبر، متناقض و نامربوط باشد. فیلتر کردن این داده‌ها ضروری است. فیلتر کردن می‌تواند تأثیر انتقال داده‌های نظارتی بر بار شبکه را کاهش دهد و عملکرد ابر را افزایش دهد. تکنیک‌های فیلتر کردن می‌توانند بر روی داده‌ها، وضعیت منابع و وضعیت محاسباتی اعمال شوند. برای هر یک از فراهم آورندگان رایانش ابری، یک میکرو سرویس نیز وظیفه جمع آوری اطلاعات و فیلتر کردن داده ها قبل از ارسال آنها به مرکز انبارش و ذخیره سازی مرکز رصد را بر عهده دارد.

۵-۳: انبارش و تجمیع داده‌ها

تجمیع داده‌ها، فرآیندی است که در آن اطلاعات جمع‌آوری می‌گردد و سپس به صورت خلاصه شده و مناسب تحلیل‌های آماری بیان می‌شود. تجمیع داده‌ها در واحد انبارش داده ها صورت می‌پذیرد و داده‌های خصوصی را ایمن می‌کند. انبارش داده‌ها می‌تواند در فواصل منظم و مبتنی بر نیاز برنامه انجام شود. تجمیع داده‌ها می‌تواند در نظارت بر ابر با استفاده از تکنیک‌های داده‌کاوی مانند خوشه‌بندی و طبقه‌بندی پیاده‌سازی شود.

۵-۴: تجزیه و تحلیل داده

پس از تجمیع داده‌ها، آن‌ها باید پردازش و تحلیل شوند تا اطلاعات مفید استخراج گردد. تحلیل داده‌ها فرآیندی است که به منظور بررسی، تبدیل و مدل‌سازی داده‌ها و در راستای کشف، اطلاعات مفید که به تصمیم‌گیری کمک می‌کند بکار گرفته می‌شود.

تحلیل داده‌ها برای بهبود عملکرد با شناسایی وضعیت فعلی منابع، پیش‌بینی وضعیت آینده و تشخیص شرایط بحرانی و غیرعادی استفاده می‌شود.

۵-۵: گزارش‌دهی و تصمیم گیری

با استفاده از تجزیه و تحلیل داده ها، گزارش کاملی از وضعیت ابر تولید خواهد شد که بصورت گرافیکی و فرمت توصیفی ارائه خواهد شد و وضعیت ابر در یک زمان خاص را نمایش می دهد. گزارش تجزیه و تحلیل به منظور انجام اقدامات کنترلی و در راستای بهبود عملکرد ابر یا حل مشکل استفاده خواهد شد. اگر منبع خاصی در شرایط بحرانی مانند بارگذاری بیش از حد یا نشت حافظه و غیره باشد، اعلان خاصی به صورت ایمیل یا زنگ هشدار برای مدیر ارسال می‌شود.

۶ – معماری پیشنهادی

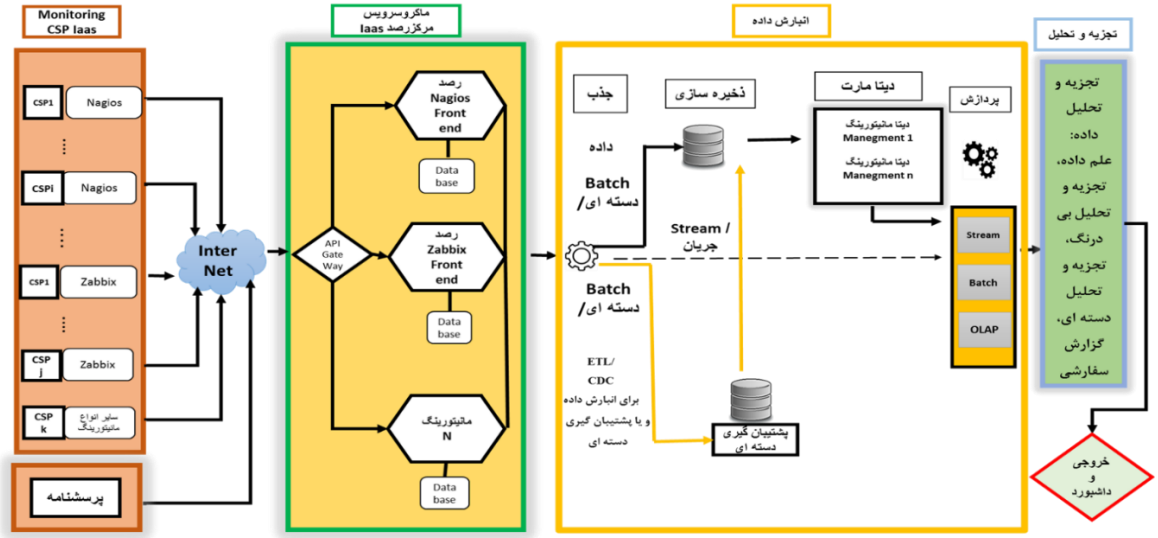
در این بخش به ارائه معماری پیشنهادی در سه سرویس (SaaS، PaaS و IaaS)[3] پرداخته می‌شود. در ابتدا استفاده از میکرو سرویس در معماری تبیین می‌گردد و در ادامه معماری نهایی مناسب سه سرویس رایانش ابری معرفی می‌شود.

۶-۱: معرفی میکروسرویس

اخیراً معماری میکروسرویس تحولات قابل توجهی در توسعه، استقرار و نگهداری مداوم برنامه‌های وب به همراه داشته است. در مقایسه با معماری سنتی برنامه‌های تکه تکه، که کل برنامه را به‌عنوان یک سیستم یکپارچه می‌سازد، مدل میکروسرویس برنامه را به چندین مؤلفه یا واحد نرم‌افزاری مستقل قابل اجرا تقسیم می‌کند که به‌طور مشترک برای ارائه عملکردهای خاص کار می‌کنند. برای تسهیل ارتباطات بلادرنگ بین میکروسرویس‌ها، روش‌هایی مانند APIهای سبک RESTful به‌طور گسترده‌ای مورد استفاده و پذیرش قرار گرفته‌اند. علاوه بر این، معماری مبتنی بر میکروسرویس به‌طور قابل‌توجهی فلسفه طراحی DevOps را با کاهش وابستگی‌های پایگاه کد منبع (Codebase) بین واحدهای نرم‌افزاری تقویت کرده است.

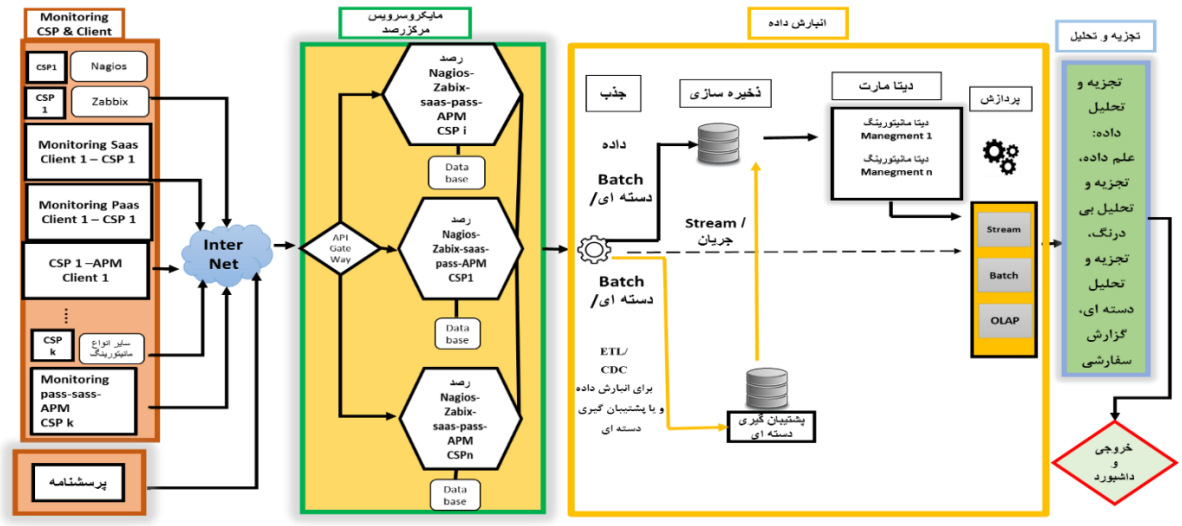
۶-۲: تشریح ساختار پیشنهادی IaaS

در شکل ۸، ساختار معماری پایش خدمت IaaS نشان داده شده است. برای پایش منابع پردازشی، انبارش داده و شبکه برای فراهم‌آورندگان خدمات رایانش ابری نرم‌افزارهای پایش Nagios، Zabbix و غیره مطرح می‌شود که این پایشگرها اطلاعات پایش خود را از طریق شبکه اینترنت برای میکرو سرویس‌های مرکز رصد ارسال می‌کنند. در مرکز رصد میکرو سرویس‌های اختصاص‌یافته به انواع مختلف پایشگرها به‌صورت اختصاصی المان‌های پایش شده برای فراهم‌کنندگان مختلف را رصد می‌کنند و در پایگاه داده محلی خود ذخیره می‌نمایند. خروجی این میکرو سرویس‌ها به‌صورت داده‌های Batch یا Stream به واحد انبارش داده انتقال می‌یابد تا ذخیره‌سازی یکسان برای آن‌ها صورت پذیرفته، پشتیبان‌گیری انجام شود و سپس به واحد پردازش و تحلیل داده Stream، Batch و OLAP انتقال یابد. مصورسازی نتایج واحد تحلیل داده به خروجی داشبورد جهت نمایش شاخص‌ها و اعلان آلارم ارسال می‌شود.



شکل ۷: معماری پایش رایانش ابری سرویس IaaS

[3] Infrastracture, Platform and Software as a Service



شکل ۸: معماری پیشنهادی رصد ترکیبی از سه سرویس SaaS, PaaS, IaaS

۶-۳: تشریح ساختار پیشنهادی SaaS و PaaS

برای پایش خدمات SaaS و PaaS ارائه شده در رایانش ابری، مرکز رصد بعنوان یک Client استفاده کننده از این خدمات به فراهم آورنده رایانش ابری، CSP، اتصال یافته و توسط نرم‌افزارهای پایش CSP SaaS PaaS شاخص‌های مربوط به کیفیت خدمات SaaS و PaaS را به میکرو سرویس موتور پایش قرار گرفته در مرکز رصد منتقل می‌کند. سپس این شاخص‌ها در پایگاه داده مربوطه تجمیع می‌گردد. لازم به ذکر است که در طرح جامع پایش IaaS این داده‌ها هنگامی‌که وضعیت منابع IaaS مناسب نباشد از اهمیت ویژه‌ای برخوردار می‌گردد. داده های مرتبط با شاخص‌های SaaS سپس به انبارش داده منتقل‌شده و پشتیبان گیری می‌شوند. بعد از آن این شاخص‌ها جهت تحلیل‌های Stream، Batch و Online به واحد پردازش ارسال می‌شوند و مصورسازی برای تحلیل‌ها انجام‌گرفته و به داشبورد مربوطه ارسال می‌شود. شکل ۹ معماری پیشنهادی ترکیبی از سه سرویسSaaS, PaaS, IaaS را نشان می‌دهد. مراحل پایش مطابق با روند پایش SaaS و IaaS و PaaS می‌باشد.

۷ – نتیجه گیری

در این مقاله، نقش پایش در سامانه های رایانش ابری تبیین شد و رویکرد های مبتنی بر عامل، غیر مبتنی بر عامل و ترکیبی معرفی گردید. در ادامه، حوزه‌های عملیات ابری تسهیل شده با نظارت از نظر فراهم آورنده سرویس و از نظر کاربر دسته بندی و تعریف شد. این حوزه ها شامل حسابداری و صورتحساب، مدیریت SLA تامین خدمات/ منابع، برنامه‌ریزی ظرفیت، مدیریت پیکربندی اطمینان از امنیت و حریم خصوصی ومدیریت خطا بودند.برای فرایند نظارت بر آنها در سیستم نظارت ابری پنج مرحله متمایز پیشنهاد گردید که عبارت بودند از: ۱) جمع‌آوری داده‌ها، ۲) فیلتر کردن داده‌ها، ۳) تجمیع داده‌ها، ۴) تحلیل داده‌ها و ۵) هشداردهی و گزارش‌دهی. برای ارائه معماری پیشنهادی مرکز رصد ابری ابتدا چارچوب مراحل سیستم نظارت بر ابر مشخص گردید و سپس مدل معماری رصد ابری برای سه نوع خدمت SaaS، IaaS و PaaS ارائه گردید. در انتها معماری رصد ترکیبی از سه سرویس IaaS، PaaS و SaaS پیشنهاد گردید.

۸- منابع

[1] Łukasz KUFEL," Tools For Distributed Systems Monitoring", Foundations of computing and decision sciences Vol. 41, No 3, 2016.

[2] Boccia V. et al., Infrastructure Monitoring for distributed Tier1: The ReCaS project use-case, International Conference on Intelligent Networking and Collaborative Systems, Salerno, Italy, 2014.

[3] Hakulinen T., Ninin P., Nunes R., Riesco-Hernandez T., Revisiting CERN Safety System Monitoring (SSM), Proceedings of International Conference on Accelerator & Large Experimental Physics Control Systems, San Francisco, California, USA, 2013.

[4] Live monitoring console of Wikimedia Grid, http://ganglia.wikimedia.org, Feb 2016.

[5] J. Gutierrez-Aguado و J. M. Alcaraz Calero, "IaaSMon: Monitoring Architecture for Public Cloud," 4 march 2016.

[6] Available online at: https://virgool.io/@elahep110/

[۷]م. حسین پور, پ. گودرزی, ن. برزگر و پ. ابوالقاسمی, نویسندگان, مشاوره و نظارت بر طراحی مرکز رصد و پایش شبکه ملی اطلاعات، ۱۴۰۰.

[8] Kufel L., Security Event Monitoring in a Distributed Systems Environment, IEEESecurity & Privacy, vol. 11, no. 1, pp. 36-43, 2013.

[9] E. Elmroth, F.G. Marquez, D. Henriksson, D.P. Ferrera, Accounting and billing for federated cloud infrastructures, in:

[24] A. Bala, I. Chana, Fault tolerance-challenges, techniques and implementation in cloud computing, Computer Science and Engineering Department, Thapar University Patiala, Punjab, India.

[25] R. Jhawar, V. Piuri, M. Santambrogio, Fault tolerance management in cloud computing: A system-level perspective, Syst. J. IEEE 7 (2) (2013) 288–297.

روش ارجاع: د.ملکی، پ.گودرزی، م.میرصراف، ی.سیفی، پیشنهاد رویکردی جامع و ترکیبی برای معماری رصد خدمات ابری، دوفصلنامه محاسبات و سامانه‌های توزیع شده، سال هفتم، شماره۱، شماره پیاپی۱۳، صفحه ۶۹ تا ۷۹، سال۱۴۰۳.

How to cite: D.Maleki, P.Goudarzi, S.M.Mirsarraf, U.Seifi, **Proposing a Comprehensive and Hybrid Approach for Cloud Service Monitoring Architecture,** Journal of Distributed Computing and Systems (JDCS), Vol 7, Issue 1, Pages 69 - 79, 2024.

داود ملکی از اعضای هیئت علمی پژوهشگاه ارتباطات و فناوری اطلاعات ایران می باشد. وی کارشناسی ارشد کامپیوتر از دانشگاه فردوسی مشهد می باشد. مقطع کارشناسی در رشته کامپیوتر به اتمام رساند. زمینه های مورد علاقه ایشان مجازي سازي، رایانش ابري، کلان داده، مراکز داده می باشد.
نشانه رایانامه ایشان عبارتست از:
dmaleki@itrc.ac.ir

پژمان گودرزی، عضو هیئت علمی و دانشیار پژوهشگاه ارتباطات و فناوری اطلاعات ایران می باشد. وی دارای دکترای مهندسی برق از دانشگاه صنعتی اصفهان بوده و در زمینه های تحقیقاتی مانند بهینه سازی منابع، جاری سازی ویدئو بر روی شبکه، شبکه های توزیع محتوا، مراکز داده و رایانش ابری فعال است.
نشانه رایانامه ایشان عبارتست از:
pgoudarzi@itrc.ac.ir

سید محمدرضا میرصراف از اعضای هیئت علمی و مشاور پژوهشگاه ارتباطات و فناوری اطلاعات ایران می باشد. ایشان دارای مدرک دکترای مخابرات از دانشگاه آزاد اسلامی واحد علوم و تحقیقات می باشد. کارشناسی و کارشناسی ارشد خود را دانشگاه خواجه نصیرالدین طوسی دریافت نموده‌اند.

Grid and Cooperative Computing, 2009 GCC'09. Eighth International Conference on, IEEE, 2009, pp. 268–275.

[10] K. Park, J. Han, J. Chung, Themis: a mutually verifiable billing system for thecloud computing environment, in: IEEE Transaction on Service Computing http://dx.doi.org/10.1109/TSC.2012.1.

[11] V. Sekar, P. Maniatis, Verifiable resource accounting for cloud computingservices, in: Proceedings of the 3rd ACM Workshop on Cloud ComputingSecurity Workshop, ACM, 2011, pp. 21–26.

[12] V.C. Emeakaroha, I. Brandic, M. Maurer, S. Dustdar, Low level metrics to high level SLAs-LoM2HiS framework: bridging the gap between monitored metrics and SLA parameters in cloud environments, in: High Performance Computing and Simulation (HPCS), 2010 International Conference on, IEEE,

[13] Z. Haiteng, S. Zhiqing, Z. Hong, Z. Jie, Establishing service level agreement requirement based on monitoring, in: Cloud and Green Computing (CGC), 2012Second International Conference on, IEEE, 2012, pp. 472–476.

[14] M. Palacios, J. Garcia-Fanjul, J. Tuya, G. Spanoudakis, Identifying test requirements by analyzing SLA guarantee terms, in: Web Services (ICWS),2012 IEEE 19th International Conference on, IEEE, 2012, pp. 351–358

[15] M. Comuzzi, C. Kotsokalis, G. Spanoudakis, R. Yahyapour, Establishing and monitoring SLAs in complex service based systems, in: Web Services, 2009.ICWS 2009. IEEE International Conference on, IEEE, 2009, pp. 783–790.

[16] A.J. Ferrer, F. Hernández, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri,R. Sirvent, J. Guitart, R.M. Badia, K. Djemame, et al., Optimis: a holistic approach to cloud service provisioning, Future Gener. Comput. Syst. 28 (1) (2012) 66–77.

[17] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, D. Pendarakis, Efficient resource provisioning in compute clouds via VM multiplexing, in: Proceedings of the 7th International Conference on Autonomic Computing, ACM, 2010, pp. 11–20.

[18] Q. Zhang, L. Cherkasova, E. Smirni, A regression-based analytic model for dynamic resource provisioning of multi-tier applications, in: Autonomic Computing, 2007. ICAC '07. Fourth International Conference on, 2007,

[19] T. Pueschel, D. Neumann, Management of cloud infrastructures: Policy-based revenue optimization, in: Thirtieth International Conference on Information Systems (ICIS 2009), 2009, pp. 1–16.

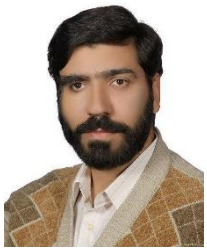
[20] D.A. Menascé, P. Ngo, Understanding cloud computing: Experimentation and capacity planning, in: Computer Measurement Group Conference, 2009.

[21] A. Sekiguchi, K. Shimada, Y. Wada, A. Ooba, R. Yoshimi, A. Matsumoto,Configuration management technology using tree structures of ICT systems,in: Proceedings of the 15th Communications and Networking Simulation Symposium, Society for Computer Simulation International, 2012, p. 4.

[22] S. Ferretti, V. Ghini, F. Panzieri, M. Pellegrini, E. Turrini, QoS aware clouds,in: Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on,IEEE, 2010, pp. 321–328.

[23] R. Jhawar, V. Piuri, M. Santambrogio, A comprehensive conceptual systemlevel approach to fault tolerance in cloud computing, in: Systems Conference (SysCon), 2012 IEEE International, IEEE, 2012, pp. 1–5.

علایق ایشـــان در زمینه رایانش ابری، هوش مصـــنوعی، امنیت سـیسـتم‌های مخابراتی، شـبکه‌های اجتماعی و شـبکه‌های ارائه محتوا و نشانه شناسی می‌باشد.
نشانه رایانامه ایشان عبارتست از:
mirsaraf@itrc.ac.ir



یونس ســیفی از اعضـــای هیئت علمی پژوهشگاه ارتباطات و فناوری اطلاعات ایران می بـاشـــد. دکترای خود را از دانشـــگاه صـــنعتی کوئینزلد اســـترالیا دریافت نموده اند. کارشناسی ارشد خود را از دانشـگاه صـنعتی شـریف دریافت کرده اند. زمینه های مورد علاقه ایشـان تحلیل و ارزیابی پروتکل، امنیت شـــبکه و ابزارهای تحلیل فرمال، شـــبکه های پتری رنگ شده می‌باشند.
نشانه رایانامه ایشان عبارتست از:
y.seifi@itrc.ac.ir

Proposing a Comprehensive and Hybrid Approach for Cloud Service Monitoring Architecture

D.Maleki[1], P.Goudarzi[2], S.M.Mirsarraf[3], U.Seifi[3]
[1] ICT Research Institute, Tehran, Iran
[2] ICT Research Institute, Tehran, Iran
[3] ICT Research Institute, Tehran, Iran
[4] ICT Research Institute, Tehran, Iran

Abstract
In this paper, the architecture of a cloud computing providers' monitoring system is first described, followed by an explanation of the observatory center designed and implemented based on this architecture. Cloud computing monitoring is essential to ensure the health, performance, security, and compliance of cloud-based services. By implementing an observatory center based on the proposed architecture's components, layers, and monitoring tools, organizations can enhance cloud resource management, respond to incidents more quickly, and maintain or improve their operational excellence. To achieve this, monitoring layers and tools for distributed systems, along with relevant APIs, have been studied and utilized for collecting cloud computing system-related data. Additionally, key areas of cloud operations have been examined and facilitated for monitoring, and ultimately, the processes related to cloud monitoring within the observatory center have been implemented.