نشریه علمی «نوآوری‌های فناوری اطلاعات و ارتباطات کاربردی»
سال اول، شماره ۱، پاییز ۱۳۹۸؛ ص ۸- ۱

بررسی چندمستاجری در محیط‌های محاسبات ابری

آرش غفوری*
دانشجوی دکتری دانشگاه علم و صنعت ایران
(دریافت: ۹۷/۱۲/۰۳؛ پذیرش: ۹۸/۰۲/۲۹)

چکیده

زیرساخت‌های فناوری اطلاعات سازمان‌ها، متحمل طیف گسترده‌ای از هزینه‌ها اعم از هزینه‌های سخت‌افزاری و نرم‌افزاری است. این در حالی است که محیط‌های محاسبات ابری با ظهور خود چشم‌انداز محسوسی برای کاهش برخی از این هزینه‌ها ارایه داده‌اند. نرم‌افزار به‌عنوان سرویس، گونه‌ای از خدمات ارایه‌شده توسط محیط محاسبات ابری است که با دیدگاهی جدید، به ارایه راهی موثر و هوشمندانه برای رفع نیاز تقاضا محور کاربر نهایی به منابع محاسباتی فراهم می‌کند. در این راستا، چندمستاجری یکی از مهمترین ویژگی/های لایه نرم‌افزار به‌عنوان سرویس است که به دلیل افزایش نرخ بهره‌وری منابع و کاهش هزینه سراسری زیرساخت‌های فناوری اطلاعات، مورد توجه سازمان‌های فناوری اطلاعات قرار گرفته است. به دلیل نو ظهور بودن این مفهوم و مزیت‌های ارایه‌شده توسط آن، بررسی ویژگی‌ها، مفاهیم وابسته و چالش‌های آن از اهمیت بالایی برخوردار است. این مقاله ابتدا به معرفی چندمستاجری، ویژگی ها و مفاهیم وابسته به آن پرداخته و سپس لایه های فنی و انواع رویکردهای مختلف مدیریت داده در معماری چندمستاجری را بررسی نموده و در انتها چالش‌های چندمستاجری و راه حل‌های موجود به منظور رفع این چالش‌ها را ارایه می‌نماید.

واژگان کلیدی

چندمستاجری، بارکاری، مقیاس‌خودکار، بهره‌وری منبع، کیفیت سرویس

۱- مقدمه

زیرساخت‌های فناوری اطلاعات سازمان‌ها، متحمل طیف گسترده‌ای از هزینه‌ها اعم از هزینه‌های سخت‌افزاری و نرم‌افزاری است. چندمستاجری یک الگوی معماری است که در آن یک نرم‌افزار سامانه‌ای میان چندین مستاجر به اشتراک گذاشته‌شده و امکان استفاده همزمان چندین مستاجر از آن نرم‌افزار فراهم می‌شود [۱]. به دلیل افزایش نرخ بهره‌وری منابع و کاهش هزینه سراسری زیرساخت‌های فناوری اطلاعات، موردتوجه سازمان‌های فناوری اطلاعات قرارگرفته است. استفاده از نرم‌افزارهای چندمستاجر علی‌رغم مزایایی همچون مقرون‌به‌صرفگی، چالش‌هایی را نیز با خود به دنبال دارد که مقیاس‌پذیری ازجمله مهم‌ترین این چالش‌ها به شمار می‌آید. وجود نوسان در تعداد درخواست و حجم منابع مورد تقاضای مستاجرها موجب ایجاد تغییرات پویا در بارکاری سامانه‌های نرم‌افزاری شده و چالش‌هایی را برای فراهم‌کنندگان ابری این سامانه‌ها به ارمغان آورده است که ازجمله آن‌ها می‌توان به اتلاف قابل‌توجه منابع محاسباتی و کاهش معیارهای کیفیت سرویس به‌خصوص زمان‌پاسخ اشاره کرد. امروزه راه‌های مختلفی برای رفع این چالش‌ها انجام‌شده است که یکی از برجسته‌ترین آنان به‌کارگیری یک مکانیزم مقیاس‌خودکار کارآمد است.

به سبب نوسان بارکاری نرم افزارهای چندمستاجری، برنامه‌ریزی ظرفیت یا منابع محاسباتی برای آن‌ها آسان نیست. ظرفیت می‌تواند بر اساس بار متوسط برنامه‌ریزی شود که این امر علی‌رغم کاهش هزینه به دلیل استفاده کمتر از منابع محاسباتی، منجر به کارایی بد سیستم نرم‌افزاری گردیده و موجب دلسردی مشتریان و تاثیر منفی درآمد خواهد شد. از سوی دیگر، اگر ظرفیت برنامه‌ریزی‌شده متناسب با بارکاری قله باشد منابع محاسباتی در بیشتر اوقات بیکار باقی خواهند ماند. مقیاس‌خودکار رویکردی است که به‌منظور غلبه بر این چالشها در محیط محاسباتی مورد استفاده قرار می‌گیرد [۳].

علی‌رغم وجود چندمستاجری در محیط‌های محاسباتی گوناگون، این معماری با عنوان دقیق "چندمستاجری" برای اولین بار در محیط محاسبات ابری معرفی گردید و امروزه اغلب ارائه‌دهندگان نرم‌افزار چندمستاجر، ارائه‌دهندگان ابری هستند که

* رایانامه نویسنده پاسخگو: krghafouri@ihu.ac.ir

به ارائه نرم‌افزار به‌عنوان سرویس می‌پردازند. فراهم‌ساز خودکار منابع‌مجازی‌شده ارائه‌شده در زیرساخت محیط محاسبات ابری، امکان پیاده‌سازی برنامه‌های‌کاربردی با افزایش و کاهش مقدار منابع استفاده‌شده در این محیط را فراهم می‌آورد. این قابلیت مقیاس‌خودکار نامیده می‌شود و هدف اصلی آن تعدیل خودکار مقیاس سامانه‌ای است که برای برآورده کردن بارکاری متفاوت با حداکثر بهره‌وری منبع و باکیفیت سرویس معین، برنامه‌های‌کاربردی را راه‌اندازی می‌کند [۴]. رویکرد مقیاس‌خودکار نیز درراه برآورده‌سازی هدف اصلی خود با چالش‌هایی رویارو است که این چالش‌ها در ادامه ذکر می‌شوند. اولین چالش که یک رویکرد مقیاس‌خودکار پیش‌بینی بارکاری است [۳]. رویکرد مقیاس‌خودکار در محیط‌های محاسباتی مختلف به‌عنوان‌مثال، یک محیط ابری دربردارنده فراهم‌سازی و آزادکردن منابع مطابق تغییرات بارکاری نرم‌افزار چندمستاجر در طی زمان است. آزادکردن منابع آسان است درصورتی‌که بدست‌آوردن منابع، نرم‌افزار چندمستاجر را متحمل سربار کارکردی می‌نماید. دومین چالش شناسایی کردن نیاز منابع درست، مطابق با بارکاری ورودی است. ازآنجایی‌که تعداد مستاجرها در هر وقفه زمانی متغیراست، تعداد درخواست‌های منابع لازم نیز متفاوت است. تعداد منابع لازم تابعی از تعداد مستاجرها، ماهیت نمونه برنامه‌کاربردی و همچنین نوع فراخوانی‌هایی است که هر مستاجر روی نمونه برنامه‌کاربردی انجام می‌دهد. منابع لازم باید به‌درستی تخمین‌زده شده و فراهم گردند. تخمین منابع می‌بایست خیلی دقیق باشد تا از فراهم‌سازی کمتر یا بیشتر از نیاز منابع و تبعات آن جلوگیری گردد. سومین چالش سیاست‌های اتخاذی سیستم به‌منظور تخصیص منبع با کمترین هزینه است.

نرم‌افزارهای چندمستاجر بر اساس نوع جایگاه اتخاذی در طیف بین اشتراک‌گذاری و انزوای کد و داده در دسته‌بندی‌های مختلف قرار می‌گیرند. آن دسته از نرم‌افزارهای چندمستاجر که مدنظر این پژوهش قرار می‌گیرد، دارای کد یکسان و داده‌های مختلف و سفارشی به‌ازای هر مستاجر می‌باشند.

ابعاد مختلفی برای مقیاس‌پذیری یافت می‌شود که الگوریتم‌ها و مکانیزم‌های مقیاس‌خودکار با در نظر گرفتن آن‌ها دست به مقیاس‌کردن می‌نمایند. از سویی دیگر معیارهای مختلفی از کیفیت سرویس در توافقنامه سطح سرویس میان مستاجرها و ارائه‌دهندگان سرویس ذکرشده و ارائه‌دهندگان سرویس موظف به تضمین می‌شوند. مکانیزم مقیاس‌خودکار ارائه‌شده در این پژوهش در بعد اندازه دست به مقیاس زده و تعداد نمونه‌های برنامه‌کاربردی و ماشین‌مجازی را افزایش یا کاهش می‌دهد و همچنین با مدنظر قراردادن معیارهای کارکردی کیفیت سرویس، جهت بهره‌وری بهتر از منابع محاسباتی واحد پردازش مرکزی یا توان پردازشی با حداقل نقض زمان‌پاسخ تلاش می‌نماید.

سازماندهی مطالب ارایه شده در این مقاله به‌صورت زیر است. در فصل دوم کارهای مرتبط و مشابه با مکانیزم ارائه‌شده مورد شرح و بررسی قرار می‌گیرد. فصل سوم به توصیف مکانیزم ارائه‌شده پرداخته و پس از ارائه معماری سیستم زمان‌اجرا نرم‌افزار چندمستاجری و مولفه‌های حاضر در آن، به صورت خاص مولفه مکانیزم مقیاس‌خودکار را مورد بررسی قرارداده و زیر مولفه‌های این مکانیزم و کارکرد هریک از آن‌ها را تشریح می‌کند. فصل چهارم به ارزیابی مکانیزم پیشنهادی پرداخته و این مکانیزم را با سایر کارهای مرتبط قابل ارزیابی در این حوزه مقایسه می‌کند؛ و درنهایت در فصل پنجم پس از جمع‌بندی پژوهش‌های انجام‌شده، پیشنهادهایی جهت کارهای آتی ارائه می‌گردد.

۱- مروری بر کارهای مرتبط

همان‌طور که در بخش‌های قبل بیان گردید، امروزه اغلب ارائه‌دهندگان نرم‌افزار چندمستاجر، ارائه‌دهندگان ابری هستند که به ارائه نرم‌افزار به‌عنوان سرویس می‌پردازند، شایان‌ذکر است که تقریباً تمامی کارهای مشابه انجام‌شده نیز در محیط محاسبات ابری ارائه و بررسی شده است. محاسبات ابری به صورت فزاینده‌ای در حال گسترده‌شدن و پیچیدگی است. یک ویژگی کلیدی محاسبات ابری کشسانی است که امکان فراهم‌سازی و آزادسازی منابع محاسباتی برحسب تقاضا با استفاده از مقیاس‌خودکار فراهم می‌کند. فن‌های مقیاس‌خودکار مختلفی یافت می‌شوند که در لایه‌های مختلف نرم‌افزار، بستر و زیرساخت مورداستفاده قرار می‌گیرند. این فن امکان فراهم‌سازی منابع محاسباتی درست مطابق با نیاز و درخواست‌های ورودی مستاجرها یا کاربران را فراهم می‌آورد [۲۸]. همه راه‌حل‌های مقیاس‌خودکار ارائه‌شده به کاربران خود این اجازه را می‌دهند که با درنظرگرفتن سرویس میزبانی‌شده در ابرها، قوانینی را تعریف نمایند. هر قانون متشکل از یک یا چندین شرط و همچنین چندین عملیات است که در صورت برآورده شدن این شرط‌ها این عملیات اجرا می‌گردند. هر یک از شرط‌ها معمولاً بر اساس مجموعه‌ای از معیارهای کیفیت‌سرویس است و به بررسی آستانه‌های تعیین‌شده برای این معیارها می‌پردازد. در صورت پیمایش هر یک از آستانه‌های تعیین‌شده در هر زمان، اقدام لازم انجام می‌شود [۳۱].

در این بخش کارهای گذشته در زمینه مقیاس‌خودکار در محیط محاسبات ابری را که نزدیکی بیشتری با کار ما دارد، مورد بررسی قرار خواهیم گرفت.

۱-۱- معیارهای ارزیابی

کارهای مرتبط انجام‌شده در حوزه مقیاس‌خودکار توسط ملاک‌های مختلف و متعددی مورد ارزیابی قرار می‌گیرند. دو معیار بسیار مهم که اغلب هم در صنعت و هم دانشگاه موردتوجه است معیارهای رویکردمقیاس و کیفیت سرویس است که در ادامه به آن اشاره شده است.

۱-۱-۱- رویکردهای مقیاس

فن‌ها و راه‌حل‌های مقیاس‌خودکار معمولاً از دو رویکرد پیروی می‌کنند که عبارت‌اند از ۱- رویکرد واکنشی[1] ۲- رویکرد پیشگیرانه[2] [۳۷].

رویکرد واکنشی دلالت بر حالتی می‌نماید که فن یا راه‌حل مقیاس‌خودکار با توجه به وضعیت جاری سیستم شروع به تصمیم‌گیری و اقدام می‌نماید. اغلب راه‌حل‌هایی مقیاس‌خودکار که توسط شرکت‌های تجاری یا متن‌باز ارائه می‌شود از این شیوه استفاده می‌نمایند.

رویکرد پیشگیرانه دلالت بر حالتی می‌نماید که در آن فن یا راه‌حل‌های مقیاس‌خودکار از فن‌های پیچیده برای پیش‌بینی تقاضای آینده استفاده کرده و به آماده‌سازی منابع مطابق با پیش‌بینی می‌پردازند. پیش‌بینی ازآن‌جهت حائز اهمیت است که همواره یک تاخیر زمانی از زمان اجرای اقدامات مقیاس‌خودکار تا تاثیر آن وجود دارد که این زمان مربوط به آماده‌سازی و راه‌اندازی منابع مورد نظر می‌شود [۳۷-۳۶].

به دلیل این‌که سامانه‌های واکنشی قادر به مقیاس‌نمودن انفجار بارکاری نیستند، به‌کارگیری فعالیت‌های پیشگیرانه به‌منظور مقیاس‌نمودن تقاضاهای درحال نوسان حائز اهمیت است.

۱-۱-۲- نوع کیفیت سرویس

فن‌ها یا راه‌حل‌های مقیاس‌خودکار می‌بایست همچنین به‌وسیله‌ی نگه‌داری از یک کیفیت سرویس قابل‌قبول، از درستی کارکرد برنامه‌کاربردی حمایت نمایند. کیفیت سرویس به دو نوع توافق‌نامه سطح سرویس وابسته است: ۱- توافق‌نامه سطح سرویس‌برنامه‌کاربردی که یک قرارداد میان مستاجر یا سرویس‌گیرنده و کاربرنهایی است. ۲- توافق‌نامه سطح سرویس منبع که میان ارائه‌دهنده منبع و مستاجر است.

اغلب تلاش‌های انجام‌شده در این حوزه از توافق‌نامه سطح سرویس نوع اول پیروی کرده و معیارهای کیفیت سرویس نوع اول مدنظر داده‌اند که به‌طور دقیق‌تر معیارهای عملکردی نامیده می‌شوند. معیارهای عملکردی خود شامل زمان‌پاسخ، بهره‌وری از منبع، توان عملیاتی و طول صف منبع هستند [۳۸].

۱-۲- کارهای مرتبط

کارهای مرتبطی که تاکنون ارائه شده‌اند را می‌توان از ابعاد گوناگون دسته‌بندی و بررسی نمود. در این مقاله کارهای مرتبط در دو دسته کارهای مرتبط دانشگاهی و کارهای مرتبط صنعتی مورد بررسی قرارگرفته‌اند. منظور از کارهای مرتبط دانشگاهی محصولات ارائه‌شده در مقالات و پایان‌نامه‌های علمی است و مقصود از کارهای مرتبط صنعتی محصولات مقیاس‌خودکار ارائه‌شده توسط سازمان و شرکت‌های بزرگ ارائه‌دهنده سرویس‌های ابری و چندمستاجری است. کارهای مرتبط صنعتی که درزمینه‌ی مقیاس‌خودکار صورت گرفته است خود به دو صورت تجاری و متن‌باز توسعه داده شده و به ارائه سرویس به کاربران یا مستاجرها می‌پردازند. کارهای مرتبط صنعتی تجاری و در برخی موارد متن‌باز که اغلب از سوی ارائه‌دهندگان سرویس مطرح ارائه می‌شود، معمولاً تنها به معرفی طرحواره یا کاربردی از سرویس خود اکتفا کرده و اطلاعات جامعی را در اختیار پژوهشگران قرار نمی‌دهند. علی‌رغم این موضوع، سرویس‌های ارائه‌شده توسط این شرکت‌ها تصویر درست و نسبتاً جامعی از اصول مقیاس‌خودکار را بازتاب نموده و بررسی چالش‌ها و نواقص آن‌ها پژوهشگران را در یافتن راه‌حلی جامع و دور از اشکال در این زمینه یاری می‌نماید. در ادامه ابتدا کارهای مرتبط دانشگاهی معرفی‌شده سپس با توجه به اطلاعات ارائه‌شده توسط خود سرویس‌دهندگان و همچنین اطلاعات کسب شده توسط برخی از مقالات به بررسی کارهای مرتبط صنعتی پرداخته می‌شود.

۱-۳- نوع کیفیت سرویس

جدول (۱): مقایسه میان چندمستاجری و چندنمونه‌ای

	مقرون به صرفگی	مقیاس پذیری	سفارشی سازی	انزوای منابع	به‌اشتراک‌گذاری منابع
چند نمونه ای	✗	✗	✓	✓	✓
چند مستاجری	✓	✓	✓	✓	✓

[1] Reactive
[2] Proactive

۲- ویژگی‌های معماری چندمستاجری

ویژگی‌های معماری چندمستاجری در حقیقت نیازمندی‌های لازم برای داشتن این نوع معماری است که شرح آن‌ها به صورت زیر می‌باشد.

۲-۱- به‌اشتراک‌گذاری منبع

هدف و انگیزه اصلی از توسعه معماری چندمستاجری صرفه جویی[1] هزینه‌های مدیریت و توسعه برای نرم‌افزارهای میزبانی‌شونده، از طریق خدمت‌رسانی به چندین مستاجر با ارایه پایگاه‌کد و منابع داده به‌اشتراک‌گذاشته شده است. نمونه نرم‌افزاری باید منابع سخت‌افزاری و نرم‌افزاری همچون زیرساخت‌سخت‌افزاری، ماشین‌های‌مجازی، سیستم‌های‌عامل، پایگاه‌های‌داده و کد را به اشتراک بگذارد. از سویی دیگر نرم‌افزار می‌بایست با استفاده از یک پایگاه‌کد واحد برای مستاجرها متعدد، توسعه را به‌اشتراک‌گذاشته و از منابع نگه‌داری کند [۱۳].

۲-۲- مقیاس‌پذیری

از آن جاییکه که مستاجرهای متعدد می‌بایست بتوانند به صورت بالقوه به هزاران مشتری سرویس بدهند، مقیاس‌پذیری[2] یکی از ویژگی معماری مهم برای معماری چندمستاجری است. مقیاس‌پذیری به توانایی سیستم در اداره کردن مقدار در حال رشد کار با یک شیوه مطلوب و همچنین به آسانی بزرگ شدن، اشاره می‌کند [۱۳]. برای مثال این بدان معنی می‌باشد که زمان پاسخ گویی برنامه‌کاربردی با افزایش بارکاری[3] (افزایش تعداد کاربرانی که به صورت همزمان از سیستم استفاده می‌کنند) پایدار باقی می‌ماند.

به سبب محدودیت‌های ذاتی افزایش مقیاس[4] (افزایش منابع برای یک گره واحد)، توانایی برای پراکندگی مقیاس[5] (افزایش تعداد گره‌ها) یک ویژگی معماری سیستم چندمستاجری است.

۲-۳- نگهداشت‌پذیری

در حالی که بسیاری از برنامه‌های کاربردی تک‌مستاجری میزبانی‌شده، برای سفارشی‌سازی مستاجرها به پایگاه‌های کد مختلف متکی هستند، معماری چندمستاجری به یک پایگاه‌کد به‌اشتراک‌گذاشته بین چندین مستاجر وابسته است. این قابلیت به کاهش تلاش برای نگه‌داری و تعمیر نرم‌افزار کمک می‌کند. به همین دلیل است که اشکالات تنها نیاز به یکبار تعمیر دارند و به روز رسانی نیز می‌تواند به صورت مرکزی راه‌اندازی شود [۱۳]. طراحی چندمستاجری با یک پایگاه‌داده اشتراک گذاشته شده موجب کاهش هزینه‌های نگه‌داری و مدیریت پایگاه‌داده برای هر یک از مستاجرها می‌شود.

۲-۴- سفارشی‌سازی

یکی دیگر از ویژگی‌های مهم معماری چندمستاجری توانایی تولید سفارشی‌سازی[6] مخصوص مستاجر است. به دلیل به‌اشتراک‌گذاشته‌شدن پایگاه‌کد برنامه‌کاربردی و پایگاه‌داده، این اهمیت دارد که به مستاجرها اجازه اتخاذ داده و منطق کسب‌وکاری متناسب با نیازمندی‌های مشتری شان داده شود [۱۳]. یک معماری چندمستاجری خوش طراحی، قادر است تعادلی بین منابع به‌اشتراک‌گذاشته شده و توانایی سفارشی‌سازی کاربر بیابد.

۲-۵- استفاده‌پذیری

علاوه بر تغییر منطق کسب‌وکار و داده یک برنامه‌کاربردی، همچنین رابط‌کاربری می‌بایست به واسطه سفارشی‌سازی مخصوص مستاجر، قابل پیکره‌بندی باشد. این به مستاجرها مختلف امکان ایجاد نشان تجاری خودشان را برای یک برنامه‌کاربردی می‌دهد.

۳- لایه‌های فنی در معماری چندمستاجری

در این بخش با معرفی دسته بندی و لایه‌های فنی چندمستاجری، به ارایه تصویری بزرگ و مفید از فنون چندمستاجری پرداخته می‌شود. سرویس‌های محیط محاسبات ابری در سه دسته تقسیم‌بندی می‌شوند. زیرساخت به‌عنوان سرویس[7]، سکو به عنوان سرویس[8] و نرم‌افزار به‌عنوان سرویس. هر چه از لایه زیرساخت به سوی لایه نرم‌افزار حرکت می‌کنیم بر میزان ارتباط و کنترل کاربر افزوده می‌شود.

پنج لایه فنی برای چندمستاجری در نظر گرفته شده است که هر کدام دارای دسته بندی مربوط به خود است [۱۲].

لایه اول، لایه ابتدای سخت‌افزار و نرم‌افزار نام دارد که بنیاد و اساس سایر لایه‌های فنی است و از روش‌های پردازش سخت‌افزار و کد برنامه‌نویسی تشکیل شده است.

لایه دوم، ابرناظر یا ناظر ماشین‌مجازی نام دارد که روش‌های این لایه با ارایه ماشین‌های مجازی جداگانه، انزوا را میان

[1] Cost Efficiency
[2] Scalability
[3] Workflow
[4] Scale Up
[5] Scale Out
[6] Customizability
[7] Infrastructure as a Service (IAAS)
[8] Platform as a Service (PAAS)

مستاجرها بر پا می‌سازند. انزوا در صورتی امکان‌پذیر می‌شود که ابر ناظر و ماشین‌مجازی کارکردهای صحیحی داشته باشند.

لایه سوم، سیستم‌عامل نام دارد که روش‌های این لایه با منزوی کردن فرآیندهای خود کاربران را منزوی می‌کند. بنابراین، سفارشی‌سازی هر انزوا برای کاربر به فرآیندها محدود است. همچنین انزوا تنها در صورت جدایی امتیاز و کارکرد صحیح هسته امن است.

لایه چهارم برنامه‌کاربردی نام دارد که روش‌های آن به‌صورت منطقی، تقسیم‌بندی انجام‌شده در یک پایگاه‌داده را منزوی می‌کنند. در این لایه سفارشی‌سازی به‌طور کامل در لایه برنامه‌کاربردی انجام می‌شود. داده‌ها می‌بایست میان کاربران منزوی شده و به‌صورت امن ذخیره شوند. برای دست‌یابی به این مهم نیاز به تصدیق و مجوزدهی به منظور فراهم آمدن یک انزوای امن است.

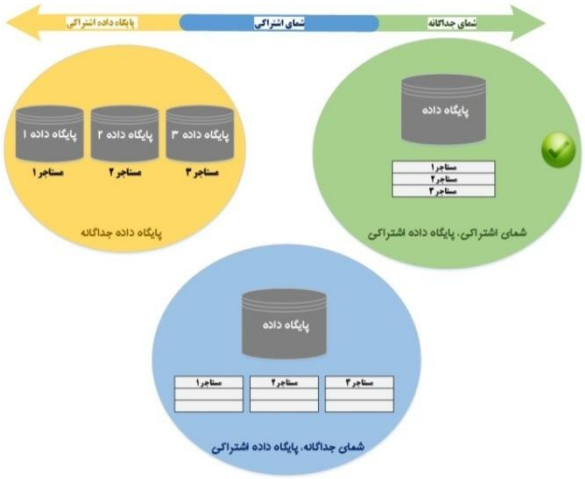
لایه پنجم، تارنما[1] نام دارد که روش‌های آن هویت کاربران را تشخیص داده و دسترسی آن‌ها را به سرویس‌های محاسبات ابری چندمستاجری روی شبکه‌ها را می‌پذیرد. هرگونه حادثه در این لایه ممکن است موجب ایجاد نقص در لایه‌های دیگر شود. جلوگیری از این امر نیازمند، امن بودن برنامه‌کاربردی طرف سرویس‌دهنده و معتبر بودن طرف مشتری است. در جدول (۲) لایه‌های فنی در معماری چندمستاجری آورده شده است.

جدول (۲): لایه‌های فنی در معماری چندمستاجری

دسته بندی	لایه‌ها
امنیت سمت سرویس‌دهنده	تارنما
امنیت سمت مشتری	
ذخیره‌سازی امن داده	برنامه‌کاربردی
انزوای داده کاربر	
تصدیق اعتبار و مجوزسازی	
جدایی امتیاز	سیستم‌عامل
یکپارچگی هسته	
امنیت ناظر ماشین‌مجازی	ابرناظر
امنیت ماشین‌مجازی	
کد برنامه‌نویسی	اولیه سخت‌افزار و نرم‌افزار
پردازش سخت‌افزار	

۴- مدیریت داده چند مستاجر

درجه انزوا برای یک برنامه‌کاربردی نرم‌افزار به عنوان سرویس می‌تواند به صورت قابل توجهی به نیازمندی‌های کسب‌وکار وابسته باشد. سه رویکرد اصلی برای مدیریت داده چند مستاجر وجود دارد که هر یک از آن‌ها در مکان مختلفی در مقیاس میان انزوا و به‌اشتراک‌گذاری قرار گرفته اند.



شکل (۱): رویکردهای مدیریت داده چندمستاجر

۴-۱- پایگاه‌داده جداگانه

در این مدل داده هر مشتری دارای یک نمونه پایگاه‌داده مستقل است. این رویکرد نسبت به سایرین دارای ارجحیت کمتر و هزینه بیشتری است. نرخ دسترسی در این مدل داده سریع و میزان انزوا در آن بسیار زیاد است [۱].

۴-۲- پایگاه‌داده مشترک- شمای جداگانه

در این مدل داده چندین مستاجر یک نمونه از پایگاه‌داده را به اشتراک می‌گذارند، در حالی که هر کدام از مستاجرها دارای شمای[2] جداگانه می‌باشند [۱و ۴]. این رویکرد از نظر ارجحیت در جایگاه متوسط قرار دارد. با توجه به توانایی، هر پایگاه‌داده می‌تواند از چندین شما نگه‌داری کند. در این رویکرد هزینه نگه‌داری و سفارش منابع کاهش می‌یابد.

۴-۳- پایگاه‌داده مشترک- شما مشترک

در این مدل داده یک شمای واحد بین مستاجرها به‌اشتراک‌گذاشته می‌شود و هر داده مستاجر بوسیله‌ی یک ستون شماره‌مستاجر، متمایز می‌شود. از مزیت‌های بسیار آشکار این مدل داده، کاهش چشمگیر هزینه است زیرا هزینه‌های منابع و نگه‌داری به وسیله ی یک توسعه بزرگ کاهش می‌یابد. امنیت و مقیاس‌پذیری داده در سطح برنامه‌کاربردی نیازمند افزایش است. این رویکرد دارای بیشترین میزان ارجحیت است و مدل داده‌ای

[1] Web

[2] Schema

است که در حال حاضر مورد استفاده بسیاری از ارایه‌دهندگان ابری قرار می‌گیرد [۱].

همان‌طور که مشاهده می‌شود هر یک از این مدل‌های داده که توسط مستاجرها انتخاب می‌شود دارای چالش‌ها می‌باشند که در ادامه به معرفی آن‌ها و راه حل‌های ارایه شده برای آن‌ها می‌پردازیم.

۵- چالش‌های چندمستاجری

در این بخش چالش‌های معماری چندمستاجری را مورد بررسی قرار گرفته و درصورت امکان مثالی برای آن در دنیای واقعی ذکر می‌شود.

۵-۱- انزوای منابع

عدم تضمین عملکرد یکی از موانع بزرگ برای کاربران بالقوه ابر است. مسائل مربوط به عملکرد اغلب برای تعداد کمی از مستاجرها که دارای بارکاری بالا هستند به وقوع می‌پیوندد. این برای یک برنامه‌کاربردی نرم‌افزار به عنوان سرویس بسیار حائز اهمیت است که با وجود به‌اشتراک‌گذاری زیرساخت و نرم‌افزار یکسان، میان منابع مستاجرها جدایی وجود داشته باشد [۱۶].

این امر طبیعی است که هر یک از مستاجرها علاقه دارند تا به منظور دست یافتن به بهترین کارایی[1] کارکرد سرویس، به تمامی منابع دسترسی پیدا کنند، که با این حال با وجود محدودیت‌های منابع این امر ممکن است برای همه مستاجرها محقق نشود. بنابراین سیستم ممکن است اولویت‌های برای مستاجرها تخصیص دهد و سرویس‌های متفاوتی را برای سرویس‌گیرنده‌های مختلف ارایه دهد. در صورت منظم یا ثابت بودن درخواست‌های سرویس گیرندگان، یکی از رویکردها تخصیص ایستای منابع همچون حافظه و پردازنده برای برنامه‌های کاربردی نرم‌افزار به عنوان سرویس است [۲]. این رویکرد در محیط محاسبات ابری بدلیل برخی از ویژگی‌های ذاتی محیط مانند تخصیص پویای منبع قابل اجرا نیست. یک مستاجر می‌تواند با ارایه نیازمندی‌های منابع خود مانند الگوی استفاده، پیش از زمان، امکان زمان بندی منابع برای برنامه‌کاربردی بدهد.

به‌عنوان مثال Google App Engine، کشش افقی خودکار[2] برای برنامه‌کاربردی ایجاد می‌کند. با این‌حال، این مخصوص مستاجر نیست و اگر کشش محدود باشد، نمی‌تواند انزوای عملکرد را تضمین کند. رویکردی توسط لین و همکارانش [۱۵] پیشنهاد شده است که به تضمین انزوای کارایی درون یک برنامه‌کاربردی چندمستاجری می‌پردازد.

۵-۲- سفارشی‌سازی

یک برنامه‌کاربردی نرم‌فزار به‌عنوان سرویس، اغلب امکان سفارشی‌سازی سرویس‌های خود را مطابق با نیازمندی‌های کیفیت سرویس می‌دهد. برای مثال، Google Doc، به کاربران خود اجازه تعیین ویژگی مختلف رابط کاربر گرافیکی و جنبه‌های طراحی (نگاه و احساس[3]) نرم‌افزار را می‌دهد و ممکن است در آینده اجازه تعیین ویژگی‌ها باتوجه به نیازمندی‌های سطح سرویس به هر کاربر داده شود. ارایه‌دهندگان نرم‌افزار به‌عنوان سرویس دیگر مانند Salesforce نیز هستند که طیف گسترده‌ای از گزینه‌ها شامل کد مخصوص مستاجر را ارایه می‌دهند.

باید توجه نمود که معماری چندمستاجری از یک نمونه نرم‌افزاری استفاده می‌کند و بنابراین، اطلاعات سفارشی‌سازی مستاجرها باید در یک پایگاه‌داده ذخیره شود. برای ارایه یک سرویس سفارشی شده نیاز به بازیابی و استفاده از این اطلاعات در زمان اجرا است. این امر در عین افزایش انعطاف‌پذیری به دلیل افزودن پیچیدگی پایگاه‌داده، ناشی از ذخیره انواع داده مربوط به نیازهای سفارشی‌سازی و محاسبات اضافی لازم در زمان اجرا، موجب کاهش سرعت پردازش می‌شود [۲]. همچنین تغییر پیکره‌بندی یک مستاجر نباید در رفتار و ظاهر برنامه‌کاربردی دیگر مستاجرها تاثیر بگذارد. یک برنامه‌کاربردی با اجازه دادن به توسعه‌های کد مخصوص کاربر قدرتمندترین راه برای اتخاذ نیازهای کاربر را فراهم می‌کند. این امر به وقوع چالش‌های فنی در چندمستاجری منجر می‌شود. میتزنر[4] و همکارانش [۱۴] یک رویکرد معماری سرویس‌گرا محور برای برنامه‌های کاربردی چندمستاجری ارایه داده‌اند که در آن یک مستاجر می‌تواند به جایگذاری یا توسعه کد بدون تاثیرگذاری بر دیگران بپردازد.

۵-۳- امنیت

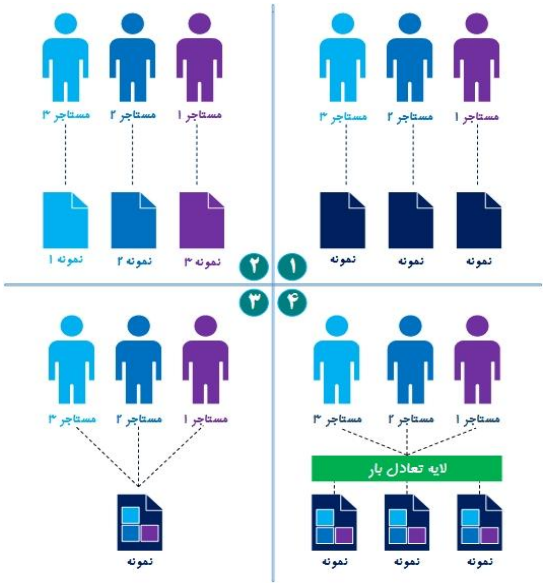
در یک طراحی چندمستاجری امنیت در لایه‌های مختلف طراحی همواره یکی از چالش‌های مهم به حساب می‌آید. این مسائل علاوه بر کد و داده نرم‌افزار به‌اشتراک‌گذاشته شده، ماشین‌های مجازی ارایه‌دهنده سرویس و ابرناظرهای آنان را نیز شامل می‌شود. از جمله چالش‌هایی که در سطح مجازی سازی سرویس‌های ارایه چندمستاجری با آن روبه‌رو هستیم وجود ماشین‌های مجازی خوابیده و مدیریت برخورد با آنان است، به

[1] Performance
[2] Automatic Horizontal Elasticity
[3] Look and Feel
[4] Mietzner

علاوه وابستگی ماشین‌های مجازی مشتریان که درحال دریافت سرویس تنها به یک ابرناظر یکی از مسائلی است که احتمال وقوع خطر امنیتی را افزایش می‌دهد [۱۸]. در چندمستاجری کد و داده نرم‌افزار میان مستاجرها به‌اشتراک‌گذاشته می‌شود و این یک خطر امنیتی قابل توجه ایجاد می‌کند. یک مستاجر به‌صورت تصادفی یا طراحی شده ممکن است به داده‌های وابسته به مستاجرها دیگر دسترسی پیدا کند. مسائل امنیتی در واقع یکی از مسائل بسیار قابل توجه در محیط محاسبات ابری و نرم‌افزار به عنوان سرویس است که هنوز راه‌حل مطلوبی برای آن ارایه نشده است.

۵-۴- مقیاس‌پذیری

یکی دیگر از چالش‌های مطرح شده در معماری چندمستاجری مقیاس‌پذیری است. یک برنامه‌کاربردی مقیاس‌پذیر باید بتواند با افزایش تعداد مستاجرها و یا افزایش منابع درخواستی توسط مستاجرها همچنان قادر به ارایه سرویس همانند گذشته باشد. چانگ و همکارانش در یک از مباحث اولیه مربوط به برنامه‌های کاربردی چندمستاجری، یک مدل بلوغ برای نرم افزار به‌عنوان سرویس پیشنهاد کرده‌اند که شماره‌های بالاتر به میزان بیشتر اشتراک گذاری در سطوح اشاره می‌کند [۹]. همچنین تنها حالت سطح ۳ و ۴ از این مدل بلوغ با توجه به تعریف چندمستاجری محسوب می‌شود.



شکل(۲): شمایی از طرح ارایه شده از چانگ

در سطح بلوغ ۱ برای هر مستاجر یک نمونه برنامه‌کاربردی در نظر گرفته شده است، در سطح بلوغ ۲ نمونه در نظر گرفته شده به ازای هر مستاجر قابلیت پیکره‌بندی را نیز دارا است. در سطح بلوغ ۳ برای کلیه مستاجرین تنها یک نمونه برنامه‌کاربردی قابل پیکره‌بندی وجود دارد که به ارایه سرویس می‌پردازد. در سطح بلوغ ۱ تا سطح بلوغ ۳، یکی از مهمترین ملاحظات مربوط به مسائل برنامه‌نویسی و طراحی نرم‌افزار است. سطح ۳ نرم‌افزار به عنوان سرویس، به نرم‌افزار اجازه استفاده توسط همه مستاجرها را می‌دهد و بنابراین، در تلاش پیاده‌سازی و طراحی نرم‌افزار به‌صورت قابل توجه صرفه‌جویی می‌شود. با این حال در سطح ۳ برنامه‌کاربردی نرم‌افزار به‌عنوان سرویس، اگر زیرساخت نسخه‌های متعدد از نرم‌افزار یکسان برای ارایه پویای سرویس نداشته باشد، ممکن است با محدودیت مقیاس‌پذیری روبه رو شود [۲]. در سطح بلوغ ۴ این مشکل نیز حل شده و در این سطح برنامه‌کاربردی چندمستاجری با افزایش مقیاس نمونه برنامه‌کاربردی موجود در سطح بلوغ ۳، به‌وسیله ارایه یک لایه توزیع‌بار حاوی تعدادی از نمونه‌های یکسان به ارایه یک نمونه برنامه‌کاربردی قابل پیکره‌بندی و مقیاس‌پذیر برای سرویس‌دهی به تعداد زیادی از مستاجرین می‌پردازد. آزیز[1] و همکارانش [۱۷] در این زمینه کار کرده و به ارایه یک رویکرد کشسان توزیع‌بار شده به منظور رسیدگی به مقیاس‌پذیری و توزیع‌بار در برنامه‌های کاربردی چندمستاجری پرداخته اند.

۶- نتیجه‌گیری

در این مقاله ابتدا تعریف روشنی از مفهوم مستاجر و چندمستاجری ارایه و تمایز آن با مفاهیم مشابه همچون چندنمونه‌ای بیان شد. سپس قابلیت‌های به‌اشتراک‌گذاری‌منبع، مقیاس‌پذیری، نگه‌داشت‌پذیری، سفارشی‌سازی و استفاده‌پذیری به عنوان ۵ ویژگی مهم معماری چندمستاجری در محیط محاسبات ابری مطرح و تشریح گردید. برای مفهوم چند مستاجری لایه‌های فنی متشکل از تارنما، برنامه‌کاربردی، سیستم‌عامل، ابرناظر و لایه اولیه سخت‌افزار و نرم‌افزار وجود دارد که به‌صورت مختصر شرح داده شد. در این معماری همچنین سه رویکرد برای مدیریت داده یافت می‌شود که عبارتند از ۱- پایگاه‌داده جداگانه، ۲- پایگاه‌داده مشترک- شمای جداگانه و ۳- پایگاه‌داده مشترک- شمای مشترک که توضیحاتی راجع به هر کدام از این رویکردها ارایه شد. معماری چندمستاجری مانند هر معماری دارای چالش‌هایی است که این چالش‌ها عبارتند از انزوای منابع، سفارشی‌سازی، امنیت و مقیاس‌پذیری، که به تفضیل به هر کدام از آن‌ها پرداخته شد و راه‌حل‌های ارایه شده برای این چالش‌ها بیان گردید.

[1] Azeez

۷- منابع

[1] S. Paliwal, "Cloud application services (SaaS) – Multi-Tenant Data Architecture", Infosys Technologies Limited, Nov, 2012.

[2] H. Koziolek, "Towards an Architectural Style for Multi-Tenant Software Applications", In Proceedings of Software Engineering, 2010, pp 81-92.

[3] Q. Shao, "Towards Effective and Intelligent Multi-tenancy SaaS", A Dissertation Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy, Jan, 2011.

[4] L. Jiang, J. Cao, P. Li and Q. Zhu, "A Mixed Multi-Tenancy Data Model and Its Migration Approach for the SaaS Application", IEEE Asia-Pacific Services Computing Conference (APSCC), Guilin, Dec, 2012

[5] B. Sengupta and A. Roychoudhury, "Engineering Multi-Tenant Software-as-a-Service Systems", Proceedings of the 3rd International Workshop on Principles of Engineering Service-Oriented Systems (ICSE), New York, USA, 2011.

[6] P. Mathur and N. Nishchal, "CloudComputing: New Challenge to The Entire Computer Industry", Conference on 1st International Parallel Distributed and Grid Computing (PDGC), Solan, Oct, 2010.

[7] C. P. Bezemer and A. Zaidman, "Challenges of Reengineering into Multitenant Saas Applications", Delft University of Technology, Netherlands, Tech. Report, TUD-SERG-2010-012, 2010.

[8] C. P. Bezemer and A. Zaidman, "Multi-Tenant SaaS Applications: Maintenance Dream or Nightmare?", Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), New York, USA, Sept, 2010.

[9] F. Chong, G. Carraro, and R. Wolter, "Multi-tenant Data Architecture", Available at:

http://msdn.microsoft.com/en-us/library/aa4790

86.aspx, 1June 2006, Last Visit At : April, 2013.

[10] T. Kwok, T. Nguyen, and L. Lam, "A Software as a service With Multi-tenancy Support for an Electronic Contract Management Application", IEEE International Conference on Services Computing (SCC), Honolulu, July, 2008.

[11] C. Bezemer, A. Zaidman, B. Platzbeecker, T. Hurkmans and A. Hart, "Enabling Multi-Tenancy",Delft University of Technology, Netherlands Tech. Report, TUD-SERG-2010-030, 2010.

[12] T. Takahashi, G. Blancy, Y. Kadobayashiy, D. Fally, H. Hazeyamay and S. Matsuo, "Enabling Secure Multitenancy in Cloud Computing: Challenges and Approaches", 2nd Baltic Congress on Future Internet Communications (BCFIC), Vilnius, April, 2012.

[13] H. Koziolek, "Towards an Architectural Style for Multitenant Software Applications", In Proceedings of Software Engineering, 2010.

[14] R. Mietzner, F. Leymann and M. P. Papazoglou, "Defining Composite Configurable SAAS Application Packages Using SCA, Variability Descriptors and Multi-Tenancy Patterns", The Third International Conference on Internet and Web Applications and Services, 2008.

[15] H. Lin1, K. Sun, S. Zhao and Y. Han, "Feedback-Control-based Performance Regulation for Multi-Tenant Applications", 15th International Conference on Parallel and Distributed Systems, 2009.

[16] A. Croll, "Bitcurrent Cloud Computing Survey: Cloud Adoption, Concerns, and Motivations", Tech. Report, March, 2011.

[17] A. Azeez, S. Perera, D. Gamage, R. Linton, P. Siriwardana, D. Leelaratne, S. Weerawarana and P. Fremantle, "Multi-Tenant SOA Middleware for Cloud Computing", 3rd International IEEE Conference on Cloud Computing (CLOUD), 2010.

[18] V. Manral, March 21, 2013, "4 Security Considerations Around Multi-Tenancy", Available at:

http://h30507.www3.hp.com/t5/HP-Networking/4-security-considerations-around-multi-tenancy/ba-p/123301#.UmOKIBC6ulo.html, Last Visit at: Oct, 2013.

Journal of "Innovations of Applied Information and Communication Technologies"
Vol. 1, No. 1, 2019 (Serial No. 1)

A Survey of Multi-Tenancy in Cloud Computing Environments

A. Ghafoori*

Iran University of Science & Technology

Abstract

Enterprise IT infrastructures endure a wide range of costs including software and hardware costs. However, the advent of cloud computing environments have provided a tangible perspective for reduce some of these costs. Software as a Service (SaaS) is type of services that provided by the cloud. It provide a new paradigm, which goal is to provide an effective and intelligent way to support end users' on-demand requirements to computing resources. In this regard, multitenancy is one of the most important features of SaaS tier that due to increasing rate of resource utilization and decreasing overall cost of IT infrastructure by IT organization have been considered. Because of the emerging nature of this concept and the advantages which is offered by it, study of characteristics, related concepts and its challenges are important. This paper first introduces Multi-tenancy, features and related concepts and then examine technical layers and variety of date management approaches in multi-tenancy architecture and finally offers multi-tenancy's challenges and existing solutions in order to resolve theirs challenges.

Keywords: Cloud Computing, SaaS, Multi-Tenancy, Resource Sharing

* Corresponding author E-mail: krghafouri@ihu.ac.ir