مجله علمی - پژوهشی "پدافند الکترونیکی و سایبری"
سال چهارم، شماره ۴، زمستان ۱۳۹۵، ص ۱۱۰-۹۹

ارائه یک مکانیزم مقیاس خودکار زمان اجرا برای نرم افزارهای چندمستاجر

آرش غفوری[۱]، محسن شریفی[۲*]

۱- کارشناسی ارشد، ۲- استاد، دانشگاه علم و صنعت ایران

(دریافت: ۹۵/۰۷/۰۵، پذیرش: ۹۵/۰۸/۱۰)

چکیده

نرم‌افزارهای چندمستاجر به دلیل نوسان بارکاری که متاثر از افزایش یا کاهش تعداد درخواست مستاجرها و حجم منبع درخواستی آن‌ها است، نسبت به مقیاس‌خودکار نمونه‌های نرم‌افزاری اقدام می‌نمایند. هدف از مقیاس‌خودکار، بهره‌وری بهتر از منابع محاسباتی همراه با ارائه کیفیت سرویس مطلوب به مستاجرها است. رویکرد مقیاس‌خودکار درراه برآورده‌سازی هدف به کار گرفته‌شده خود با چالش‌هایی روبه‌رو است که این چالش‌ها متشکل از پیش‌بینی بارکاری، شناسایی کردن نیاز منابع درست مطابق بارکاری و چگونگی سیاست اتخاذی سیستم به‌منظور تخصیص منبع با کمترین هزینه است. در این مقاله با ارائه یک مکانیزم مقیاس‌خودکار زمان‌اجرا برای نرم‌افزارهای چندمستاجر راهکاری برای پاسخگویی به چالش‌های ذکرشده ارائه شده است. در ابتدا به دلیل عدم وجود تصویری جامع از سیستم زمان‌اجرا یک نرم‌افزار چندمستاجر، با توجه به شواهد یافت‌شده حاصل از بررسی‌ها و مطالعات گوناگون، به ارائه یک معماری برای سیستم زمان‌اجرا نرم‌افزار چندمستاجر پرداخته شده و مدل رایانش با سه دانه‌بندی برای مولفه‌ی نرم‌افزار چندمستاجر آن معرفی می‌گردد. سپس در ادامه با ارائه یک معماری برای مولفه مقیاس‌خودکار سیستم زمان‌اجرا نرم‌افزارهای چندمستاجر و به‌کارگیری زیرمولفه‌هایی با کارکردهای مناسب، جهت پاسخگویی به چالش‌های ذکرشده برای مقیاس‌خودکار اقدام می‌شود. به‌منظور ارزیابی، مکانیزم مقیاس‌خودکار ارائه‌شده در محیط شبیه‌ساز کلادسیم پیاده‌سازی شده و با جدیدترین کار مشابه در شرایط کاملاً یکسان مورد آزمایش و مقایسه قرار می‌گیرد.

واژه‌های 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] و 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 از جمله آن‌ها می‌توان به اتلاف قابل توجه 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.

به‌سبب 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 اوج 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 [۲].

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

* رایانامه نویسنده مسئول: msharifi@iust.ac.ir

Archive of SID
www.SID.ir

معین، برنامه‌های کاربردی را راه‌اندازی می‌کند [۳]. رویکرد مقیاس‌خودکار نیز در راه برآورده‌سازی هدف اصلی خود با چالش‌هایی روبه‌رو است که این چالش‌ها در ادامه ذکر می‌شوند. اولین چالش که یک رویکرد مقیاس‌خودکار پیش‌بینی بارکاری است [۲]. 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 به‌دست‌آوردن 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 به ازای 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 آن 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 با حداقل نقض زمان‌پاسخ تلاش می‌نماید.

سازماندهی مطالب ارایه‌شده در این مقاله به صورت زیر است: در فصل دوم، کارهای مرتبط و مشابه با مکانیزم ارائه‌شده، مورد شرح و بررسی قرار می‌گیرد. فصل سوم، معماری پیشنهادشده برای سیستم زمان‌اجرا نرم‌افزار چندمستاجر و مولفه‌های حاضر در آن را معرفی می‌نماید و در فصل چهارم، مولفه مکانیزم مقیاس‌خودکار به صورت خاص مورد بررسی قرار گرفته و زیر مولفه‌های این مکانیزم و کارکرد هریک از آن‌ها را تشریح می‌گردد. فصل پنجم به ارزیابی مکانیزم پیشنهادی پرداخته و این مکانیزم را با سایر کارهای مرتبط قابل ارزیابی در این حوزه مقایسه می‌کند. درنهایت در فصل ششم پس از جمع‌بندی پژوهش‌های انجام‌شده، پیشنهادهایی جهت کارهای آتی ارائه می‌گردد.

۲- مروری بر کارهای مرتبط

به سبب آن‌که عنوان دقیق "چندمستاجری" برای اولین بار در محیط محاسبات ابری[1] معرفی گردیده است و همچنین امروزه اغلب ارائه‌دهندگان نرم‌افزار چندمستاجر، ارائه‌دهندگان ابری هستند که به ارائه نرم‌افزار به‌عنوان سرویس می‌پردازند، تقریباً تمامی کارهای مشابه انجام‌شده نیز در محیط محاسبات ابری ارائه شده است. یک ویژگی کلیدی محاسبات ابری کشسانی است که امکان فراهم‌سازی و آزادسازی منابع محاسباتی را 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 [۴]. 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 [۵].

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.

۲-۱- معیارهای ارزیابی

کارهای مرتبط انجام‌شده در حوزه مقیاس‌خودکار توسط ملاک‌های مختلف و متعددی مورد ارزیابی قرار می‌گیرند. دو معیار بسیار مهم که در اغلب کارهای مرتبط ارائه‌شده در صنعت و دانشگاه مورد توجه است، رویکردمقیاس و کیفیت سرویس است.

1- Cloud Computing

Archive of SID

www.SID.ir

۲-۱-۱- رویکردهای مقیاس

فن‌ها و راه‌حل‌های مقیاس‌خودکار معمولاً از دو رویکرد پیروی می‌کنند که عبارت‌اند از ۱- رویکرد واکنشی ۲- رویکرد پیشگیرانه [۶].

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 از آن‌جهت حائز اهمیت است که همواره یک تاخیر زمانی از زمان اجرای اقدامات مقیاس‌خودکار تا تاثیر آن وجود دارد که این زمان جهت آماده‌سازی و راه‌اندازی منابع مورد نظر است [۷-۶].

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

۲-۱-۲- 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 [۸].

۲-۲- 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.

۲-۲-۱- کارهای مرتبط دانشگاهی

منظور از کارهای مرتبط دانشگاهی محصولات ارائه‌شده در مقالات و پایان‌نامه‌های علمی است که با ارائه الگوریتم یا محصولی راه‌حل‌هایی را به‌منظور مقیاس‌خودکار معرفی نموده‌اند. اولین کار مرتبط دانشگاهی توسط آقای هان و همکاران [۹] ارائه شده است. الگوریتم پیشنهادی توسط آقای هان و همکاران به صورت واکنشی است که با مدنظر قراردادن معیار زمان‌پاسخ، به مقیاس‌خودکار منابع زیرساخت پرداخته و جهت بهبود بهره‌وری منابع اقدام می‌نماید. به دلیل واکنشی بودن این الگوریتم مقیاس‌خودکار هر بار به‌محض نقض زمان‌پاسخ انجام‌شده و این به‌کرات منجر به نقض توافق‌نامه سطح سرویس و تحمیل هزینه جریمه حاصل از آن می‌شود. از سویی دیگر مقیاس‌خودکار توسط این الگوریتم نیازمند برخی ملاحظات است که وجود آن‌ها همیشه امکان‌پذیر نیست. به‌عنوان‌مثال به‌منظور انجام مقیاس خوددرمانی می‌بایست ماشین‌مجازی با منبع بیکار وجود داشته باشد و یا به‌منظور مقیاس سطح منبع می‌بایست همیشه قسمتی از منابع ارائه‌دهندگان فیزیکی از قبل رزرو شده باشند که خود این موضوع نیز هزینه اضافی را به ارمغان می‌آورد و در انتها مقیاس سطح ماشین‌مجازی تنها در صورت داشتن منابع کافی از سوی ارائه‌دهنده فیزیکی امکان‌پذیر می‌شود.

در کاری دیگر آقای کجراوال [۱۰] به ارائه فن‌های جدیدی به‌منظور بهینه‌سازی بهره‌وری عملیاتی در ابرها می‌پردازد و از رویکرد مقیاس پیشگیرانه بهره‌گیری کرده است. معیاری که توسط قوانین الگوریتم‌ها مورد نظارت قرار می‌گیرد تعداد درخواست‌های کاربران در ثانیه است. ازاین‌رو می‌توان میزان بهره‌وری از منابع را به‌عنوان معیار اصلی آن عنوان کرد. در الگوریتم‌های یادشده تعداد گره‌هایی که به ازای هر بار اقدام مقیاس‌خودکار افزوده یا کاسته می‌شود به‌عنوان ورودی ذکرشده و معیار معینی برای چگونگی تعیین تعداد گره‌ها برای این ورودی ذکر نشده است. از سویی دیگر به‌منظور انجام مقیاس‌خودکار پیشگیرانه، آستانه‌هایی تعیین می‌شود که برابر با یک مقدار ثابتی از ظرفیت نهایی موجود از منابع است. با توجه به وجود احتمال نوسان در میزان تقاضا، تعیین حد بالایی برای منابع محاسباتی با در نظر گرفتن مقادیر ثابت تجربی گاها موجب ایجاد محدودیت و عدم بهره‌وری درست از منابع محاسباتی می‌شود.

و در آخرین کار مرتبط دانشگاهی که توسط آقای اقبال حسین [۱۱] ارائه شده است به طراحی، پیاده‌سازی، انتشار و ارزیابی معماری‌های ابر-محور بسیار مقیاس‌پذیر که به‌منظور تکامل سریع و توان‌بالا برای کسب‌وکار و همچنین به‌منظور ملاقات نیازمندی‌های برنامه‌های‌کاربردی وب‌محور ایشان پرداخته شده است. این اثر با هدف غلبه بر چالش‌های ماهیت غیرقابل

Archive of SID

www.SID.ir

پیش‌بینی ترافیک‌های تقاضای کاربران برنامه‌های کاربردی و درنتیجه منابع محاسباتی مورد تقاضای آن‌ها، یک مکانیزم مقیاس خودکار ارائه کرده است. در مکانیزم مقیاس خودکار ارائه شده، به‌منظور تعیین مقدار آستانه مناسب برای معیارهای کیفیت‌سرویس اتخاذی و عدم نقض توافق‌نامه سطح سرویس از الگوریتم پیشنهادی کجراوال استفاده شده است و تمامی موارد تحلیلی ذکرشده در بالا در مورد این اثر نیز صدق می‌کند.

۲-۲-۲- کارهای مرتبط صنعتی

مقصود از کارهای مرتبط صنعتی محصولات مقیاس خودکار 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 میان شرکت‌های بزرگ تجاری و ابری، آمازون، میکروسافت و رایت‌اسکیل از برجسته‌ترین شرکت‌های کامپیوتری می‌باشند که به ارائه راه‌حل‌های مقیاس‌خودکار می‌پردازند. در میان شرکت متن باز ابری نیز شرکت‌های اکالیپتوس، اسکیل‌آر و کلادیا به‌منظور ارائه سرویس‌مقیاس‌خودکار نسبت به توسعه و یا استفاده از مولفه‌های مقیاس‌خودکار اقدام کرده‌اند. نتیجه بررسی شرکت‌های یادشده به این حقیقت اشاره دارد که، کارهای مرتبط صنعتی ارائه‌شده از روش‌های قوانین‌محور واکنشی استفاده می‌کنند. در این روش‌ها شرایط مقیاس‌کردن روی یک معیار هدف (به صورت اختیاری و یا اجباری) و با آستانه‌ای مشخص تعیین می‌شود و به‌محض رسیدن به آستانه‌ی تعیین‌شده توسط کاربرها و یا شرکت‌های سرویس‌دهنده دست به مقیاس‌خودکار می‌زنند [۱۲]. فراتر از آستانه‌های ثابت تعیین‌شده [۱۳]، آنان از یک روش رگرسیونی نیز جهت اتخاذ پویای آستانه‌ها برای ملاقات اهداف کیفیت سرویس استفاده می‌کنند. از مهم‌ترین چالش‌هایی که در این نوع کارها به چشم می‌خورد عدم پیش‌بینی بارکاری یا درخواست ورودی آینده به‌منظور بهره‌وری بهتر از منابع محاسباتی با کمترین نقض معیارهای کیفیت سرویس ارائه‌شده در توافق‌نامه سطح سرویس است

در جدول (۱) کارهای مرتبط انجام شده در دسته بندی‌های مختلف و همراه با مقادیر اتخاذشده توسط آنان برای انواع معیارهای ارزیابی یادشده، به نمایش درآمده است.

جدول (۱). دسته‌بندی ارائه‌شده برای کارهای مرتبط

انواع کارهای مرتبط			نوع رویکرد مقیاس	نوع کیفیت سرویس
کارهای مرتبط دانشگاهی		مکانیزم مقیاس‌پویا مبتنی بر الگوریتم کجراوال	پیشگیرانه	بهره‌وری از منبع و زمان‌پاسخ
		فن‌هایی برای ردپای ابری در حال بهینه	پیشگیرانه	زمان‌پاسخ
		مقیاس‌نمودن منابع سبک‌وزن برای برنامه‌های‌کاربردی ابری	واکنشی	بهره‌وری از منبع
کارهای مرتبط صنعتی	تجاری	ابر محاسباتی کشسان آمازون	واکنشی	بهره‌وری از منبع
		میکروسافت آژر	واکنشی	بهره‌وری از منبع
		رایت‌اسکیل	واکنشی	بهره‌وری از منبع
	متن‌باز	اکالیپتوس	واکنشی	بهره‌وری از منبع
		اسکیل آر	واکنشی	بهره‌وری از منبع
		کلادیا	واکنشی	بهره‌وری از منبع

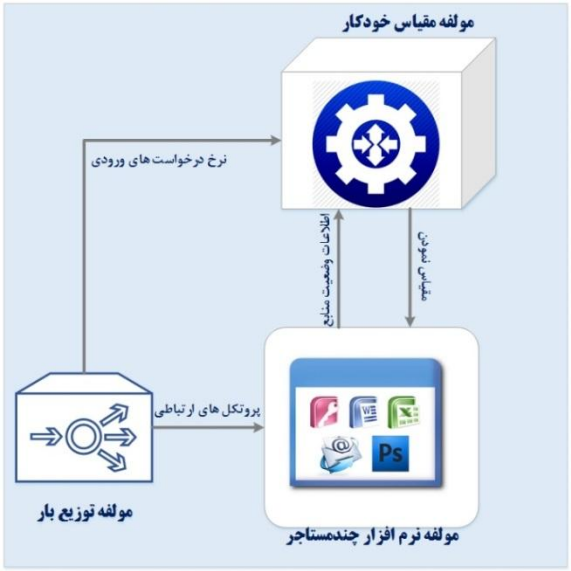
۳- معماری سیستم زمان اجرا نرم افزار چندمستاجر

معماری سیستم زمان‌اجرا که در این پژوهش برای سیستم‌نرم‌افزار چندمستاجر ارائه‌شده است متشکل از سه مولفه‌ی اصلی با نام‌های مولفه توزیع‌بار، مولفه مقیاس‌خودکار و مولفه‌ی نرم‌افزار چندمستاجر است. هریک از این مولفه‌ها وظیفه یا مجموعه وظایفی را بر عهده دارد که متناسب با نوع وظیفه خود به تعامل با سایر مولفه‌های دیگر سیستم می‌پردازند. شکل (۱) مولفه‌ها و نوع تعاملات هریک از آن‌ها را به نمایش می‌گذارد. در

Archive of SID

www.SID.ir

ادامه هر یک از مولفه‌های ذکرشده در معماری سیستم چندمستاجر معرفی و مولفه مقیاس‌خودکار به صورت کامل تشریح می‌گردد.



شکل (۱). معماری سیستم زمان‌اجرا نرم‌افزار چندمستاجر

۳-۱- مولفه نرم‌افزار چندمستاجر

همان‌طور که در گذشته ذکر شد، نرم‌افزار چندمستاجر نرم‌افزاری است که از معماری چندمستاجری پیروی کرده و نوع کاربرد ارائه‌شده را به صورت همزمان میان چندین مستاجر به اشتراک می‌گذارد. برای نرم‌افزار چندمستاجر یک مدل رایانش با سه سطح دانه‌بندی در نظر گرفته‌شده است که در ادامه به شرح آن خواهیم پرداخت.

۳-۱-۱- سطح‌های دانه‌بندی مدل رایانش نرم‌افزار چندمستاجر

موجودیت‌های مدل رایانش معرفی‌شده برای نرم‌افزارهای چندمستاجر مطابق شکل (۲) خود متشکل از سه سطح دانه‌بندی است که این سطح‌ها به شرح زیر هستند.

۳-۱-۱-۱- نمونه برنامه کاربردی

سطح اول دربردارنده نمونه‌های برنامه‌کاربردی است که به صورت همزمان به ارائه یک خدمت به چندین سرویس‌گیرنده و یا مشتری می‌پردازد. نمونه برنامه‌کاربردی خود می‌تواند متشکل از یک یا مجموعه‌ای از سرویس‌ها باشد که می‌توانند به صورت مرتبط یا مستقل از یکدیگر به ارائه خدمت می‌پردازند.

۳-۱-۱-۲- نمونه ماشین مجازی

سطح دوم دربردارنده نمونه‌های ماشین‌مجازی است که با بهره‌گیری از فناوری مجازی‌سازی و با ارائه یک ماشین‌مجازی روی ناظر ماشین‌مجازی، محیطی با توان زیرساختی معین و متاثر از میزان منابع فیزیکی لایه زیرین، برای اجرای نمونه‌های نرم‌افزاری ارائه می‌دهد. ازآنجایی‌که هر ماشین‌مجازی دارای پیکره‌بندی زیرساختی معین است تنها امکان اجرای تعداد محدودی از نمونه‌های برنامه‌کاربردی روی آن فراهم است.

۳-۱-۱-۳- نمونه ماشین فیزیکی

سطح سوم دربردارنده نمونه‌های ماشین‌فیزیکی است. منظور از ماشین‌فیزیکی، همان سرورها و زیرساخت‌های سخت‌افزاری است که غالباً توان زیرساختی آنان با استفاده از فناوری مجازی‌سازی به ماشین‌های مجازی تبدیل می‌شود. معمولاً متناسب با میزان ظرفیت زیرساختی هر ماشین‌فیزیکی، چندین ماشین‌مجازی روی آن در حال اجرا است.

شکل (۲). مدل‌رایانش نرم‌افزار چندمستاجر

۳-۲- مولفه توزیع بار

مولفه توزیع‌بار ارائه‌شده در معماری سیستم زمان‌اجرا چندمستاجری، مولفه نرم‌افزاری است که ابتدا درگاهی را که کاربران یا مستاجرها به‌منظوردست‌یابی و دریافت سرویس مورد نظر به آن متصل می‌شوند را شنود می‌کند. سپس این مولفه درخواست‌های مستاجرها را گرفته و با توجه به نوع سیاست توزیع‌بار یا الگوریتم مربوطه آنان را برای نمونه یا نمونه‌های برنامه‌کاربردی ارسال می‌نماید.

۳-۳- مولفه مقیاس خودکار

یکی دیگر از مولفه‌های معماری پیشنهادی برای سیستم

www.SID.ir

زمان‌اجرا نرم‌افزارهای چندمستاجر، مکانیزم مقیاس‌خودکار است. هدف اصلی این مقاله ارائه یک مکانیزم مقیاس‌خودکار زمان‌اجرا برای نرم‌افزارهای چندمستاجر است که در بخش‌های قبل دلایل و ضرورت انجام آن بیان گردید.

مکانیزم مقیاس‌خودکار به‌منظور ایفای نقش درست و مقیاس‌نمودن در سطوح مختلف مولفه نرم‌افزار چندمستاجر که با هدف استفاده مقرون به صرفه از منابع همراه با حداقل نقض کیفیت سرویس انجام می‌شود، نیاز به تعامل با سایر مولفه‌های معماری سیستم نرم‌افزار چندمستاجر دارد. شرح این تعاملات بدین گونه است. ابتدا نرخ ورود و یا تعداد درخواست‌های ورودی به سیستم به منظور دریافت سرویس، توسط این مولفه و از مولفه توزیع‌بار دریافت می‌شود. این مولفه علاوه بر مولفه توزیع بار همزمان با مولفه نرم‌افزار چندمستاجر نیز تعامل دارد. این تعامل دو طرفه است بدین صورت که معمولاً اطلاعات لازم از وضعیت منابع محاسباتی و نمونه‌های برنامه کاربردی، ماشین‌مجازی و ماشین‌فیزیکی توسط مکانیزم مقیاس‌خودکار دریافت می‌شود و با توجه به اطلاعات رسیده از مولفه‌های توزیع‌کننده‌بار و نرم‌افزار چندمستاجر، عملیات مقیاس‌خودکار توسط مکانیزم مقیاس‌خودکار و روی نرم‌افزار چندمستاجر انجام می‌شود. مکانیزم مقیاس‌خودکار برحسب، نیاز و با هدف حداکثر بهروری از منابع موجود و همچنین حداقل اتلاف منابع، نسبت به مقیاس‌خودکار (افزایش یا کاهش نمونه‌ها) دانه‌بندی‌های مختلف در مدل‌رایانش نرم‌افزار چندمستاجر اقدام می‌نماید.

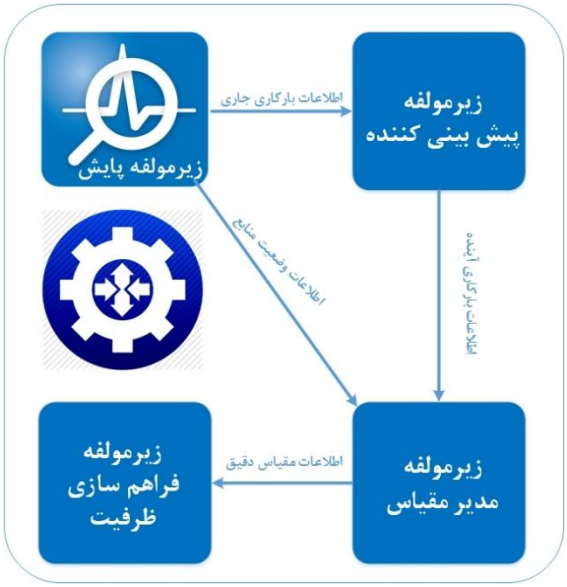
۴- معماری مکانیزم مقیاس خودکار

مکانیزم مقیاس‌خودکار که خود به‌عنوان یک مولفه در معماری سیستم زمان‌اجرا نرم‌افزارهای چندمستاجر به ایفای نقش می‌پردازد، دربردارنده‌ی چهار زیرمولفه است که هرکدام بخشی از فرآیند مقیاس‌خودکار را عهده‌دار هستند. در این بخش معماری مکانیزم مقیاس‌خودکار معرفی‌شده و سپس مولفه‌های تشکیل‌دهنده و تعاملات میان آن‌ها را به صورت کامل تشریح می‌شوند.

مولفه مکانیزم مقیاس‌خودکار متشکل از چهار زیر مولفه یا موجودیت است که عبارت‌اند از:

۱- زیرمولفه پایش
۲- زیرمولفه پیش‌بینی‌کننده
۳- زیرمولفه مدیر مقیاس
۴- زیرمولفه فراهم‌سازی ظرفیت

در شکل (۳) زیرمولفه‌های مربوط به مکانیزم مقیاس خودکار و تعاملات میان آن‌ها به نمایش داده شده است.



شکل (۳). معماری مولفه مقیاس خودکار

۴-۱- زیرمولفه پایش

این زیرمولفه وظیفه پایش و نظارت بر دو مولفه اصلی توزیع‌کننده‌بار و مولفه نرم‌افزار چندمستاجر را برعهده دارد. این زیرمولفه با پایش و نظارت بر نرخ ورودی تعداد درخواست مستاجرها به مولفه توزیع‌بار اطلاعات لازم در مورد بارکاری ورودی بر نرم‌افزار چندمستاجر را استخراج کرده و در اختیار زیرمولفه‌ی پیش‌بینی‌کننده قرار می‌دهد و از سوی دیگر این مولفه با پایش وضعیت تعداد نمونه‌ها در دانه‌بندی مختلف ارائه شده برای نرم‌افزار چندمستاجر، اطلاعات مفیدی را از وضعیت منابع محاسباتی موجود در دانه‌بندی‌های مختلف به زیرمولفه مدیرمقیاس ارائه می‌دهد.

زیرمولفه پایش اولین زیرمولفه مکانیزم مقیاس‌خودکار است که به ایفای نقش می‌پردازد. این مولفه در هریک وقفه‌های زمانی که دارای اندازه‌های برابر بوده و با $t_{interrupt}$ نمایش داده می‌شود، مولفه توزیع‌کننده‌بار را پایش می‌نماید. درنتیجه این پایش اطلاعات متعددی در مورد بارکاری سیستم ازجمله تعداد درخواست‌های سرویس ورودی مستاجرها و میزان متوسط مصرف برخی از منابع، ازجمله توان پردازشی توسط آن‌ها به مولفه پیش‌بینی‌کننده گزارش می‌شود.

نحوه محاسبه میزان مصرف منابع توسط این مولفه به این شرح است. با توجه به این حقیقت که میزان مصرف منابع محاسباتی در نرم‌افزارهای مختلف متفاوت است، امیدریاضی میزان توان‌پردازشی(p) مصرفی به ازای چندین درخواست ورودی با زمان‌پاسخ معین(t)، در تعداد آزمایش معین اندازه‌گیری می‌شود. با توجه به برخی از اطلاعات بارکاری ازجمله تعداد درخواست ورودی در هر وقفه زمانی($t_{interrupt}$)، میزان منابع

Archive of SID

www.SID.ir

لازم برای هر میزان بارکاری مطابق جدول (۲) محاسبه می‌گردد.

جدول (۲). محاسبه میزان بارکاری به‌وسیله زیرمولفه پایش مکانیزم مقیاس‌خودکار

معرفی متغیرها:

$E_t(X) = X$ امید ریاضی مصرف منبع

$\mu =$ حداکثر زمان‌پاسخ قابل‌قبول

$N_R =$ تعداد درخواست سرویس ورودی توسط مستاجرها

$W_X =$ میزان منبع مصرفی توسط نوع منبع

قدم اول: محاسبه مقادیر امیدریاضی حاصل از n بار آزمایش برای واحد پردازش مرکزی p

$$E_t(X) = \left\{ \sum_{i=0}^{n} x_i p_X(x_i) \,|\forall X \rightarrow X \in \{p\}, 0 < t \leq \mu \right\}$$

قدم دوم: محاسبه مقادیر بارکاری W بر اساس میزان واحد پردازش‌مرکزی مصرفی p در وقفه زمانی $t_{interrupt}$

$$W_X = \{N_R \times E_t(X) | \forall X \rightarrow X \in \{p\}\}$$

۴-۲- زیرمولفه پیش‌بینی‌کننده

این زیرمولفه اطلاعات استخراجی در مورد نرخ ورود درخواست مستاجرها را از زیرمولفه پایش دریافت کرده و با بهره‌گیری از مدل‌های ریاضی به پیش‌بینی پرداخته و از تحلیل مدل‌های آماری به‌وسیله سری‌های زمانی استفاده می‌کند. این زیرمولفه از این طریق نسبت به پیش‌بینی بارکاری ورودی یا نرخ ورودی درخواست مستاجرها در آینده، اقدام می‌نماید.

رویکرد مقیاس‌خودکار در نرم‌افزارهای چندمستاجر همواره درگیر فراهم‌سازی و آزادکردن منابع در قالب نمونه‌های برنامه‌کاربردی و نمونه‌های ماشین‌مجازی است که میزان آن متناسب با تغییرات بارکاری یا به عبارتی تعداد تقاضای سرویس مستاجرها در واحد زمان است. این فراهم‌سازی و آزادکردن منابع می‌بایست به صورت صحیح انجام گیرد تا با فراهم‌سازی بیش از نیاز منابع موجب افزایش هزینه منابع مصرفی و یا با آزادکردن ناصحیح و برنامه‌ریزی نشده منابع موجب کاهش کیفیت سرویس ارائه‌شده و نارضایتی مستاجرها نگردد. از سویی دیگر به‌رغم آسان بودن آزادکردن، فراهم‌سازی نمونه‌های برنامه‌کاربردی و نمونه‌های ماشین‌مجازی خود سربارهای عملکردی برای نرم‌افزار چندمستاجر به ارمغان می‌آورد. یکی از این سربارهای عملکردی که در این پروژه به‌طور خاص مورد بررسی است سربار افزایش زمان‌پاسخ است که افزایش آن از حد مجاز تعیین‌شده در توافق‌نامه سطح سرویس بین مستاجرها و ارائه‌دهنده سرویس علاوه‌بر نارضایتی مشتری، سرویس‌دهنده نرم‌افزار چندمستاجر را متحمل جریمه نیز می‌گرداند. راه‌حل مطلوب به‌منظور جلوگیری از سربارها، به‌خصوص سربار افزایش زمان‌پاسخ و همچنین به حداقل رساندن هزینه جریمه نقض کارکرد درج‌شده درتوافق نامه سطح سرویس و هزینه فراهم‌سازی منابع، نیاز به تخصیص منابع در قالب تکثیر نمونه‌های برنامه‌کاربردی و ماشین‌مجازی پیش از فرارسیدن موعد اوج نیاز منابع است. این مهم تنها از طریق پیش‌بینی بارکاری آینده و به‌وسیله‌ی داده‌های تاریخی، امکان‌پذیر است. راه‌حل ما بارکاری سیستم را از ابتدا تا وقفه زمانی فعلی دریافت و با استفاده از آن بارکاری را در وقفه زمانی بعدی پیش‌بینی می‌کند. در این استراتژی ما بارکاری سیستم که شامل میزان تقاضای توان پردازشی توسط درخواست‌های ورودی است را در وقفه‌های زمانی مختلف ثبت و سپس با توجه به داده‌های تاریخی به‌دست‌آمده، به پیش‌بینی میزان بارکاری در وقفه زمانی بعد اقدام می‌کنیم. فن‌های مختلفی برای پیش‌بینی بارکاری در ریاضیات آماری وجود دارد و ما با بررسی و بر اساس دقت پیش‌بینی روش ARIMA[1] را انتخاب نمودیم که بر اساس شواهد به‌دست‌آمده یکی از دقیق‌ترین پیش‌بینی‌کننده‌ها برای سری‌های زمانی است و به دلیل نزدیکی مقدار پیش‌بینی‌شده با مقدار واقعی خطای پیش‌بینی بسیارکم و قابل‌قبولی ارائه می‌دهد.

۴-۲-۱- پیش‌بینی بارکاری ورودی با استفاده ARIMA

فرض می‌کنیم سری زمانی $W^r_{t_{interrupt}}$ کل بارکاری ناشی از درخواست‌های ورودی باشد که میزان مصرف منبع r در وقفه زمانی $t_{interrupt}$ را نشان می‌دهد. در ادامه ما در معادلات زیر برای راحتی آن را با $W_{t_{interrupt}}$ نمایش می‌دهیم.

در پیش‌بینی یک‌قدم ما با دانستن n بار از مشاهدات قبلی $W_{t_{interrupt}}$ قصد داریم تا $W_{t_{interrupt+1}}$ که میزان استفاده از منبع در زمان $t_{interrupt} + 1$ است را در زمان $t_{interrupt}$ پیش‌بینی کنیم. اگر سری زمانی $W_{t_{interrupt}}$ از یک مدل $ARIMA(n, q)$ پیروی می‌کند و آن‌گاه در صورت ایستا بودن و برای هر $t_{interrupt}$ معادله شماره یک مدل ARIMA به شرح زیر است:

$$W_{t_{interrupt+1}} = \emptyset_0 W_{t_{interrupt}} + \cdots + \emptyset_{n-1} W_{t_{interrupt-n+1}} + \epsilon_{k+1} + \theta_0 \epsilon_k + \cdots + \theta_{q-1} \epsilon_{k+1-q} \quad (۱)$$

که در آن $\emptyset_i$ و θ_j ثابت‌های تخمین‌زده شده از داده موجود هستند. اصطلاح ϵ_k خطاهایی را به نمایش می‌گذارد که به صورت مستقل، مشابه مثال‌های توزیع‌شده از یک توزیع نرمال با میانگین صفر و واریانس محدود σ^2 فرض شده‌اند. پارامترهای n و q به

1 - Autoregressive Integrated Moving Average

Archive of SID

www.SID.ir

ترتیب تعداد وقفه‌های استفاده‌شده به‌وسیله‌ی مدل و تعداد اصطلاحات خطا هستند. اولین معادله ARIMA به صورت خلاصه به صورت زیر نیز قابل‌نمایش است:

$$W_{t_{interrupt+1}} = \sum_{i=0}^{n-1} \emptyset_0 L^i W_{t_{interrupt}} + \epsilon_{k+1} + \sum_{i=0}^{q-1} \theta_i L^i \quad (۲)$$

که در آن، L عملگر انتقال به عقب است که مطابق زیر تعریف می‌شود:

$$L^i W_{t_{interrupt}} = W_{t_{interrupt-i}} \quad (۳)$$

هرگاه، $n = 0$ و $q = 0$ باشد. حالت و یا به عبارت بهتر مدل‌های خاصی از ARIMA تحت عنوان AR و MA حاصل می‌شود. روش برازشی مدل ARIMA فرض را بر ایستا بودن داده‌ها قرار می‌دهد. درصورتی‌که سری‌های زمانی دارای نوسان باشند ما از عملگر تفاضل به‌منظور ایستا نمودن آن استفاده می‌کنیم که مطابق زیر تعریف شده است:

$$(1 - L)W_{t_{interrupt}} = W_{t_{interrupt}} - W_{t_{interrupt-1}} \quad (۴)$$

از این طریق می‌توان معادلات چندجمله‌ای از درجه $t_{interrupt}$ را با $t_{interrupt}$ بار تفاضل و با استفاده از عملگر $(1 - L)^{t_{interrupt}} y(t)$ به صورت ثابت درآورد. یک مدل ARIMA(n, d, q) یک مدل ARIMA(n, q) است که d بار تفاضل یافته است.

در پیش‌بینی چندقدمی ما قصد داریم استفاده از منابع محاسباتی را در پنجره زمانی $H \in \mathbb{N}^+$ پیش‌بینی کنیم و این امر مستلزم پیش‌بینی استفاده منابع در مرحله $h \in \mathbb{N}^+$ است به‌طوری‌که رابطه $1 \ll h \ll H$ برقرار باشد.

فرض کنید $W_{t_{interrupt}+h|t_{interrupt}}$، h امین قدم پیش‌بینی باشد که n قدم قبلی را در نظر گرفته باشد، بنابراین ما قصد پیش‌بینی $W_{t_{interrupt}+1|t_{interrupt}}, \dots, W_{t_{interrupt}+h|t_{interrupt}}$ را داریم. پیش‌بینی چند قدم با تکرار کردن چندین پیش‌بینی یک‌قدم انجام می‌شود.

۴-۳- زیرمولفه مدیر مقیاس

این مولفه نقش کلیدی را در امر مقیاس‌خودکار ایفا می‌نماید و با توجه به اطلاعات دریافتی تصمیمات لازم جهت انجام مقیاس‌خودکار را گرفته و تعداد نمونه‌های لازم به‌منظور پاسخگویی به درخواست‌های ورودی مستاجرها را تعیین می‌کند. این مولفه دو نوع اطلاعات ورودی و از دو زیر مولفه مختلف دریافت می‌نماید. اطلاعات ورودی اول مربوط وضعیت جاری منابع محاسباتی موجود در نرم‌افزار چدمستاجر است که از سوی زیرمولفه پایش برای زیرمولفه مدیرمقیاس فرستاده می‌شود. اطلاعات ورودی دوم مربوط به بارکاری پیش‌بینی‌شده توسط زیرمولفه پیش‌بینی‌کننده است که با توجه به وضعیت بارکاری جاری و بارکاری در زمان‌های گذشته دست به پیش‌بینی تعداد درخواست ورودی مستاجرها در آینده می‌زند. این مولفه با توجه به هر دو اطلاعات دست به تصمیم‌گیری می‌زند. در این بخش ما به صورت رسمی، چگونگی کارکرد مقیاس‌خودکار را شرح می‌دهیم. فرض می‌کنیم تعداد نمونه ماشین‌های فیزیکی، نمونه ماشین‌های مجازی و نمونه برنامه‌کاربردی در زمان t به ترتیب شامل $PMI_t \in \mathbb{N}^+$ نمونه ماشین‌فیزیکی، $VMI_t \in \mathbb{N}^+$ نمونه ماشین‌مجازی و $AI_t \in \mathbb{N}^+$ نمونه برنامه‌کاربردی همگون باشد. ما فرض می‌کنیم انواع منابع ارائه‌شده r برای یک ماشین‌فیزیکی، ماشین‌مجازی و نمونه برنامه‌کاربردی تنها شامل واحد پردازش مرکزی یا توان پردازشی است که آن را با $CPU \in r$ نمایش می‌دهند و همچنین برای نمایش ظرفیت کلی نوع منبع r یک نمونه و همچنین ظرفیت مصرفی آن منبع در زمان t به ترتیب از $C_t^{r,I}, C^{r,I} \in \mathbb{R}^+$ استفاده می‌کنیم. برای مدل‌سازی پویای سیستم، ما زمان را به وقفه‌های زمانی برابر با طول‌های یکسان تقسیم می‌کنیم. ما فرض می‌کنیم پیکره‌بندی مجدد سیستم از نظر تعداد انواع نمونه‌ها و در لایه‌های مختلف مدل رایانش، در هر وقفه زمانی $t_{interrupt} \in \mathbb{N}^+$ اتفاق می‌افتد. در هر وقفه میزان استفاده نوع منبع r توسط نمونه‌های مختلف اندازه‌گیری یا پیش‌بینی‌شده و با $W_{t_{interrupt}}^r$ نمایش داده می‌شود. فرض می‌کنیم $x_{t_{interrupt}} \in \mathbb{R}^+$ تعداد نمونه فعال و $u_{t_{interrupt}} \in \mathbb{R}$ تعداد تغییرات انواع نمونه‌های فعال را نشان می‌دهد. هر مکانیزم مقیاس‌خودکار دارای دو عمل اصلی افزایش و کاهش مقیاس است که در این مکانیزم پیشنهادی مبنای افزایش یا کاهش مقیاس علامت $u_{t_{interrupt}}$ است. عدد مثبت برای $u_{t_{interrupt}}$ به معنای تعداد ماشین‌های بیشتری است که به‌منظور استفاده راه‌اندازی یا روشن خواهند شد درحالی‌که مقدار منفی آن به معنای تعداد ماشین‌های فعالی است که آزاد خواهند شد. برای درک بهتر موضوع مثال ساده‌ای ذکرشده است که تعداد ماشین‌های فعال را در زمان $t_{interrupt+1}$ توصیف می‌کند:

$$x^I_{t_{interrupt+1}} = x^I_{t_{interrupt}} + u^I_{t_{interrupt}} \quad (۶)$$

شایان‌ذکر است که رویکرد اتخاذی برای انجام مقیاس‌خودکار به این‌ترتیب است که افزایش‌مقیاس را به صورت حریصانه و کاهش‌مقیاس را به صورت محتاطانه انجام می‌دهد این بدان معنی است که هنگام پیش‌بینی میزان تقاضا پیش از وقوع رویداد، درصورتی‌که پیش‌بینی افزایش میزان تقاضا را نشان دهد بلافاصله عمل افزایش مقیاس صورت می‌گیرد ولی در صورت

Archive of SID

www.SID.ir

پیش‌بینی عکس تا زمان وقوع دقیق و کاهش میزان تقاضا کاهش‌مقیاس صورت نمی‌گیرد. هدف مکانیزم ما کنترل تعداد نمونه‌های ماشین‌مجازی و برنامه‌کاربردی برای کاهش هزینه‌های سیستم به‌خصوص در هزینه مصرف منابع محاسباتی شامل توان پردازشی و همچنین تا حد امکان هزینه نقض توافق‌نامه سطح سرویس با ارائه سرویس با زمان‌پاسخ مناسب است. در همین راستا این مکانیزم مقیاس‌خودکار با تخصیص درست نمونه‌های ماشین‌مجازی و برنامه‌کاربردی باگذشت زمان و متناسب با تغییرات بارکاری و تعداد درخواست‌های سرویس موجب به حداقل رساندن تعداد نمونه‌های برنامه‌کاربردی و ماشین‌مجازی بیکار و درنتیجه حداقل‌سازی میزان مصرف منبع و هزینه‌های متاثر از آن می‌شود. از سویی دیگر، ازآنجایی‌که مقیاس‌خودکار خود به دلیل افزودن و راه‌اندازی‌کردن تعدادی از نمونه‌های برنامه‌کاربردی و ماشین‌مجازی نیازمند اندکی زمان است مکانیزم مقیاس‌خودکار با فراهم‌سازی معین نمونه‌های برنامه‌کاربردی و ماشین‌مجازی پیش از موعد تقاضا موجب کاهش سربار زمانی اضافی و کاهش زمان‌پاسخ به مستاجرها شده و هزینه ناشی از جریمه نقض توافقنامه سرویس را به حداقل ممکن می‌رساند.

همان‌طورکه در بخش‌های قبل ذکر گردید، نحوه مدیریت مقیاس‌کردن تعداد نمونه‌ها در انواع مختلف ماشین‌مجازی و نمونه برنامه‌کاربردی بر عهده مولفه مدیر مقیاس و همچنین تصمیم‌گیری راجع به چگونگی و کیفیت فراهم‌سازی آن بر عهده مولفه فراهم‌سازی ظرفیت است که در ادامه به آن می پردازیم.

بعد از این که میزان بارکاری برای زمان یا زمان‌های بعد از زمان جاری توسط مولفه پیش‌بینی‌کننده، پیش‌بینی و محاسبه شد این اطلاعات به‌منظور بررسی و تصمیم‌گیری به مولفه مدیر مقیاس داده می‌شود. این مولفه در وهله اول تعداد نمونه برنامه‌کاربردی و تعداد نمونه‌های لازم برای پاسخگویی به بارکاری را توسط تابع زیر محاسبه و با تعداد نمونه‌های برنامه‌کاربردی یا نمونه‌های ماشین‌مجازی جاری مقایسه می‌کند و تعداد نمونه‌های اضافی یا کم را به ازای بارکاری جاری محاسبه می‌کند.

$$f_1\left(W^r_{t_{interrupt+1}}\right) = x^{AI}_{t_{interrupt+1}} = \frac{W^r_{t_{interrupt+1}}}{C^{r,AI}} \quad (۷)$$

$$f_2\left(W^r_{t_{interrupt+1}}\right) = x^{VMI}_{t_{interrupt+1}} = \frac{W^r_{t_{interrupt+1}}}{C^{r,VMI}} \quad (۸)$$

حال با توجه به محاسبه $u^I_{t_{interrupt}}$ که نشانگر تعداد ماشین افزایش یا کاهش یافته میان دو وقفه کاری متوالی است فرمان مقیاس‌خودکار از سوی مدیر مقیاس صادر گردیده و جهت اعمال و انجام درست آن، اطلاعات لازم به مولفه فراهم‌سازی ظرفیت فرستاده می‌گردد. الگوریتم انجام عملیات مقیاس خودکار توسط مولفه مدیر مقیاس به صورت شبه کد در جدول (۳) نمایش داده شده است.

جدول (۳). الگوریتم انجام عملیات مقیاس‌خودکار به‌وسیله مولفه مدیرمقیاس

ورودی:

$u^{AI}_{t_{interrupt}}$: تغییرات تعداد نمونه برنامه‌کاربردی فعال در وقفه زمانی $t_{interrupt}$

$u^{VMI}_{t_{interrupt}}$: تغییرات تعداد نمونه ماشین‌مجازی فعال در وقفه زمانی $t_{interrupt}$

خروجی:

VMI_Addjustment: متغیر شرطی اعلان کاهش مقیاس نمونه ماشین‌مجازی در وقفه زمانی بعدی

AI_Addjustment: متغیر شرطی اعلان کاهش مقیاس نمونه برنامه‌کاربردی در وقفه زمانی بعدی

```
1   Begin:
2   If (VMI_Addjustment = true)
3                { Deprovision(u^VMI_tinterrupt);
        VMI_Addjustment ← false ; }
4   If (AI_Addjustment = true)
5                { Deprovision(u^AI_tinterrupt);
        AI_Addjustment ← false ;  }
6    Switch u^AI_tinterrupt
7        Case 1 (u^AI_tinterrupt > 0):
8              If (u^VMI_tinterrupt > 0)
                 {
9                  Provision (u^VMI_tinterrupt);
10                 Provision (u^AI_tinterrupt);
11               } else if (u^VMI_tinterrupt = 0)
                     {
12                       Provision (u^AI_tinterrupt);
                     }
13             Break;
14       Case 2 (u^AI_tinterrupt = 0):
15             Break;
16       Case 3 (u^AI_tinterrupt < 0):
17             If (u^VMI_tinterrupt < 0)
               {
18                   VMI_Addjustment ← true;
19                   AI_Addjustment ← true;
20               } else { AI_Addjustment ← true }
21             Break;
22             Return VMI_Addjustment, AI_Addjustment;
23   End;
```

Archive of SID

www.SID.ir

۴-۴- زیرمولفه فراهم‌سازی ظرفیت

این زیرمولفه وظیفه پیاده‌سازی تصمیم اتخاذی از سوی مدیر مقیاس را دارد و با توجه به اطلاعات دریافتی و بررسی وضعیت حاکم بر نمونه‌ها در لایه‌های مختلف مدل رایانش مولفه نرم‌افزار چندمستاجر، نسبت به افزایش یا کاهش مناسب نمونه‌های برنامه‌کاربردی و یا ماشین‌مجازی اقدام می‌نماید.

پیاده‌سازی عملیات مقیاس‌خودکار بعد از انجام تصمیم‌گیری توسط مولفه مدیرمقیاس و با فراخوانی دو تابع Provision و Deprovision توسط مولفه فراهم‌سازی ظرفیت صورت می‌گیرد. درصورتی‌که $u^{I}_{t_{interrupt}}$ برابر با یک عدد منفی باشد با فراخوانی شدن تابع Deprovision، بر اساس نوع وضعیت نمونه‌های ماشین‌مجازی و توزیع نمونه‌های برنامه‌کاربردی روی آنان و از طریق آزادکردن ماشین‌های مجازی بیکار یا با بهره‌وری کم، نسبت به عملیات کاهش مقیاس اقدام می‌گردد.

از سویی دیگر درصورتی‌که $u^{I}_{t_{interrupt}}$ برابر با یک عدد مثبت باشد با فراخوانی تابع Provision، بر اساس نوع ورودی عملیات افزایش‌مقیاس با افزودن نمونه‌های برنامه‌کاربردی و ماشین‌مجازی در لایه‌های مربوطه انجام می‌شود.

درصورتی‌که $u^{I}_{t_{interrupt}}$ برابر با یک عدد مثبت باشد این بدان معنا است که تعداد نمونه‌های نوع I می‌بایست افزایش یابد؛ بنابراین فرمان افزایش‌مقیاس توسط مدیر مقیاس و با فراخوانی تابع Provision صادر می‌گردد. پیاده‌سازی این امر توسط زیرمولفه فراهم‌سازی ظرفیت خود با توجه وضعیت جاری انواع نمونه‌ها در لایه‌های مختلف حالات مختلفی را موجب می‌گردد.

۵- ارزیابی روش پیشنهادی

در این بخش ابتدا چگونگی شبیه سازی میان افزار ارائه شده را شرح می‌دهیم. در ادامه ضمن معرفی معیارهای ارزیابی، به ارزیابی میان افزار ارائه شده می‌پردازیم.

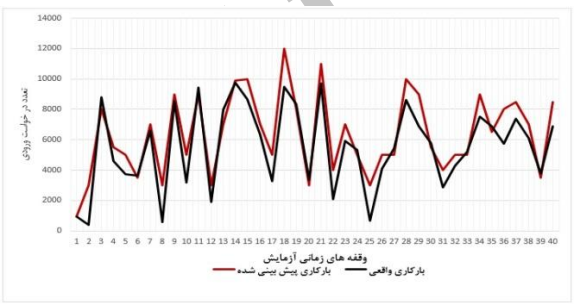
۵-۱- شیوه ارزیابی

به منظور ارزیابی، مکانیزم مقیاس‌خودکار پیشنهادی را در شبیه‌ساز کلادسیم پیاده‌سازی نمودیم. برای اثبات تحقق یافتن اهداف ذکرشده، مکانیزم مقیاس‌خودکار پیشنهادی را با مکانیزم مقیاس‌خودکار ارائه شده توسط آقای اقبال حسین که از جهت اهداف و کارکرد نیز بیشترین نزدیکی را با مکانیزم پیشنهادی دارد، در شرایط برابر و با داده های یکسان مورد آزمایش قرارداده و نتایج حاصل را ارزیابی نمودیم.

نرم‌افزارهای چندمستاجر با توجه به نوع ارائه سرویس معمولاً دارای تعداد درخواست سرویس‌های نوسانی و بارکاری متفاوت در طی زمان هستند. به همین جهت برای آزمودن و ارزیابی ۱۰ بارکاری به صورت تصادفی تولید نمودیم. برای هر بارکاری ۴۰ وقفه زمانی در نظر گرفته‌ایم که مقادیر تولیدشده در هر وقفه زمانی، تعداد درخواست‌های ورودی برای دریافت سرویس و یا بارکاری واردشده را نشان می‌دهد. هدف از تولید ۱۰ بارکاری مقایسه نتایج حاصل از اجرای مکانیزم پیشنهادی با کار مشابه انجام‌شده در چندین آزمایش گوناگون است تا بدین‌وسیله ارزیابی درست‌تری صورت گیرد. در شکل (۴) به‌عنوان نمونه شمایی از آزمایش اول انجام‌شده روی مکانیزم پیشنهادی، در قالب نمودار به نمایش درآمده است. این نمودار دربردارنده بارکاری واقعی و بارکاری پیش‌بینی‌شده توسط مکانیزم پیشنهادی در هر وقفه زمانی و به‌منظور فراهم‌سازی منابع است

شکل (۴). شمایی از آزمایش اول روی مکانیزم مقیاس‌خودکار پیشنهادی

شکل (۵) نیز شمایی از آزمایش اول انجام روی کار مشابه مقیاس‌خودکار پیشنهادی شده‌است که در قالب نمودار به نمایش درآمده و دربردارنده بارکاری واقعی و بارکاری پیش‌بینی‌شده توسط کار مشابه مکانیزم پیشنهادی در هر وقفه زمانی و به‌منظور فراهم‌سازی منابع است.



شکل (۵). شمایی از آزمایش اول روی کار مشابه با مکانیزم پیشنهادی

۵-۲- معیارهای ارزیابی

جهت ارزیابی میزان بهره‌وری از منابع محاسباتی در مکانیزم پیشنهادی، تعداد نمونه‌های بیکار موجود در سطح برنامه‌کاربردی، به‌منظور محاسبه میزان توان پردازشی مازاد مصرف، به ازای هر بارکاری وارده، مورد محاسبه و بررسی گردید. زیرا علی‌رغم وجود سطوح دانه‌بندی‌های مختلف برای مولفه نرم‌افزار چندمستاجر،

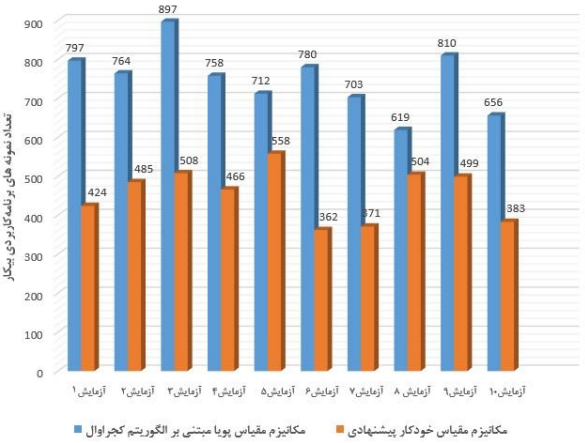
Archive of SID
www.SID.ir

درخواست‌های کاربران درنهایت توسط نمونه‌های برنامه‌کاربردی دریافت و سرویس‌دهی می‌شوند.

به‌علاوه ازآنجایی‌که پاسخگویی به بارکاری جاری می‌بایست با هدف به حداقل رساندن دفعات نقض کیفیت سرویس باشد تعداد دفعات نقض زمان‌پاسخ نیز محاسبه می‌گردد. در اینجا علی‌رغم وجود انواع مختلفی از منابع سامانه‌ای تنها منابع پردازشی یعنی واحدپردازش‌مرکزی و در میان انواع ویژگی‌های کیفیت سرویس تنها کارکرد و به صورت بسیار دقیق زمان‌پاسخ مدنظر است. در شبیه‌سازی انجام‌شده، تعداد دفعات نقض کیفیت سرویس برابر با تعداد دفعاتی است که بارکاری یا تعداد درخواست‌های سرویس واردشده به سیستم نرم‌افزار چندمستاجر بیشتر از ظرفیت قابل پشتیبانی توسط تعداد نمونه‌های برنامه‌کاربردی موجود باشد.

۵-۳- نتایج شبیه سازی و ارزیابی

به‌منظور مقایسه، نتایج حاصل از آزمایش‌های متعدد و گوناگون روی مکانیزم مقیاس‌خودکار پیشنهادی و کار مشابه، در یک نمودار ستونی کنار هم قرارداده‌شده است. در شکل (۶) تعداد نمونه‌های برنامه‌کاربردی بیکار و بدون استفاده در طی آزمایش‌های متعدد برای مکانیزم مقیاس‌خودکار و کار مشابه نمایش داده‌شده است.



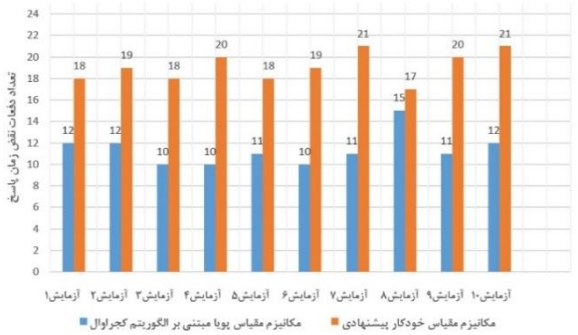
شکل (۶). مقایسه تعداد نمونه‌های برنامه‌کاربردی بیکار میان مکانیزم پیشنهادی و کار مشابه

همان‌طور که در شکل (۶) مشاهده می‌شود تعداد نمونه‌های برنامه‌کاربردی بیکار و یا به عبارت بهتر اتلاف توان‌پردازشی در مکانیزم مقیاس‌خودکار پیشنهادی کمتر از مکانیزم مشابه بوده و تفاوت چشمگیری با آن دارد.

به منظور تحلیل دقیق نتایج حاصل از مقایسه نسبت به محاسبه فاصله اطمینان اختلاف میزان توان پردازشی مصرف شده توسط مکانیزم مقیاس‌خودکار پیشنهادی با کارمشابه اقدام نمودیم.

با در نظر گرفتن میزان توان پردازشی اتلافی نسبت به کل میزان توابع پردازشی درخواستی مستاجرها برای مکانیزم پیشنهادی و کارمشابه انجام شده در نسبت‌های جداگانه و همچنین محاسبه فاصله اطمینان اختلاف نسبت آن‌ها می توان نتیجه گرفت که مکانیزم پیشنهادی با ۹۹ درصد اطمینان دارای (۱۳/۲۸-۱۶/۱۹) درصد بهره‌وری در استفاده از توان پردازشی وهمچنین دارای اتلاف کمتر توان پردازشی نسبت به کار مشابه انجام شده است.

یکی از اهداف دیگر مکانیزم مقیاس‌خودکار پیشنهادی به حداقل رساندن تعداد دفعات نقض کیفیت سرویس در عین افزایش بهره‌وری و کاهش اتلاف منابع محاسباتی است. در همین راستا و به‌منظور ارزیابی از این جنبه نیز، تعداد دفعات نقض زمان پاسخ در مکانیزم مقیاس‌خودکار پیشنهادی و کار مشابه اندازه‌گیری شده و به صورت نمودار ستونی به شرح شکل (۷) به نمایش درآمده است.



شکل (۷). مقایسه تعداد دفعات نقض زمان‌پاسخ در مکانیزم پیشنهادی و کار مشابه

همان‌طور که از نتایج به نمایش درآمده در شکل (۷) پیداست، تعداد دفعات نقض زمان‌پاسخ در مکانیزم مقیاس‌خودکار پیشنهادی نسبت به مکانیزم مقیاس‌پویا مبتنی بر الگوریتم کجراوال مقدار اندکی رشد داشته است. تعداد حداقل و حداکثر دفعات نقض زمان‌پاسخ در مکانیزم پیشنهادی به‌طور متوسط ۵ بار بیشتر از مکانیزم دیگر است که البته این به معنای افزایش هزینه نقض کیفیت سرویس مکانیزم پیشنهادی نسبت به کار مشابه نیست. دلیلی که موجب افزایش تعداد دفعات نقض زمان‌پاسخ در مکانیزم پیشنهادی نسبت به مکانیزم مشابه می‌شود مقادیر پیش‌بینی‌شده توسط پیش‌بینی‌کننده با استفاده مقادیر سری زمانی است. مقادیر پیش‌بینی‌شده گاها به‌رغم نزدیکی به مقدار بارکاری ورودی در وقفه زمانی مورد نظر، اندکی بیشتر و یا اندکی کمتر است. در حالت اول، درصورتی‌که میزان منابع

Archive of SID

www.SID.ir

پیش‌بینی‌شده بیشتر از میزان منابع لازم برای بارکاری ورودی باشد به اندازه تفاوت میان مقدار فراهم‌شده و مقدار موردنیاز، مقدار منابع بی‌استفاده یافته می‌شود. در حالت دوم، درصورتی‌که میزان منابع پیش‌بینی‌شده کمتر از میزان منابع لازم برای بارکاری ورودی باشد، در آن صورت به دلیل تخصیص بارکاری بیشتر از ظرفیت به منابع موجود زمان‌پاسخ متناسب با حجم بارکاری مازاد تحت تاثیر قرارگرفته و بیشتر می‌شود. همان‌طور که پیداست وقوع حالت دوم موجب رشد اندک دفعات نقض زمان‌پاسخ در مکانیزم پیشنهادی نسبت به کار مشابه شده است.

۶- نتیجه‌گیری

ما در این مقاله با ارائه یک مکانیزم مقیاس‌خودکار زمان‌اجرا برای نرم‌افزارهای چندمستاجر، موجب بهره‌وری بیشتر از منابع محاسباتی شده و راه‌کاری مقرون به صرفه برای پاسخگویی به بارکاری نوسانی مستاجرها ارائه نمودیم. بدین منظور، ابتدا به دلیل عدم وجود تصویری جامع از سیستم زمان اجرا یک نرم‌افزار چندمستاجر، با توجه به شواهد یافت‌شده حاصل از بررسی‌ها و مطالعات گوناگون، به ارائه یک معماری برای سیستم زمان اجرا نرم‌افزار چندمستاجر پرداختیم و یک مدل رایانش با سه دانه‌بندی برای مولفه نرم‌افزار چندمستاجر آن معرفی نمودیم. سپس با ارائه یک معماری برای مولفه‌مقیاس‌خودکار سیستم زمان‌اجرا نرم‌افزارهای چندمستاجر و به‌کارگیری زیرمولفه‌هایی با کارکردهای مناسب، در جهت پاسخگویی به چالش‌های مقیاس‌خودکار در نرم‌افزارهای چندمستاجر اقدام نمودیم.

حوزه مورد بحث در این مقاله به سبب وسعت و تازگی دارای چالش‌های متعددی است. موضوعاتی همچون چندمستاجری و مقیاس‌خودکار هرکدام به‌تنهایی به دلیل نو بودن و همچنین پیچیدگی دارای چالش‌ها و موارد قابل بررسی مختلفی هستند. سعی برآن شده است تا به تعدادی از این چالش‌ها در این پژوهش پاسخ داده شود اما هنوز چالش‌ها و مسائل متعددی وجود دارد که می‌بایست در پژوهش‌های آتی مورد بحث قرار گیرد. مکانیزم مقیاس‌خودکار ارائه‌شده تنها بر اساس میزان مصرف واحد پردازش مرکزی دست به تکثیر یا فراهم‌سازی و حذف نمونه‌ها در دانه‌بندی‌های مختلف زده است که امر می‌تواند در مورد سایر انواع منابع محاسباتی همچون حافظه، دیسک به صورت واحد یا مجتمع بررسی و نتایج تحلیل گردد. علاوه بر این معیارکیفیت سرویس مورد پایش توسط مکانیزم مقیاس‌خودکار به‌منظور میزان نقض کیفیت سرویس تنها زمان‌پاسخ بوده است که می‌تواند برای سایر معیارهای کیفیت سرویس همچون توان‌خروجی، دسترسی‌پذیری بررسی و تحلیل گردد.

۷- مراجع

[1] Q. Shao, "Towards Effective and Intelligent Multi-Tenancy SaaS," Ph.D. Thesis, Arizona State University Tempe, AZ, USA, Jan 2011.

[2] N. Roy, A. Dubey, and A. Gokhale, "Efficient Autoscaling in the Cloud Using Predictive Models for Workload Forecasting," IEEE International Conference on Cloud Computing (CLOUD), Washington DC, USA, July 2011.

[3] N. M. Calcavecchia, B. A. Caprarescu, E. Di Nitto, D. J. Dubois, and Dana Petcu, "DEPAS: a Decentralized. Probabilistic Algorithm for Auto-Scaling," Computing 94, Sep. 2012.

[4] H. Alipour, L. Yan, and H. Abdelwahab, "Analyzing Auto-scaling Issues in Cloud Environments," 2014.

[5] C. Eddy, L. Rodero-Merino, F. Desprez, and A. Muresan, "Auto-scaling, Load Balancing and Monitoring in Commercial and Open-Source Clouds," 2012.

[6] T. Lorido-Botran, J. Miguel-Alonso, and J. A. Lozano, "A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments," Journal of Grid Computing, Springer 2014.

[7] M. Mao and M. Humphrey, "A Performance Study on the VM Startup Time in the Cloud," The Fifth International Conference on Cloud Computing, IEEE Computer Society, Washington, DC, USA, 2012.

[8] F. Benevenuto, C. Fernandes, M. Santos, V. Almeida, J. Almeida, G. J. Janakiraman, and J. R. Santos, "Performance models for virtualized applications," Workshops In Frontiers of High Performance Computing and Networking, Springer Berlin Heidelberg, Jan 2006.

[9] R. Han, L. Guo, M. M. Ghanem, and Y. Guo, "Lightweight resource scaling for cloud applications," 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2012.

[10] A. Kejariwal, "Techniques for Optimizing Cloud Footprint," IEEE International Conference on Cloud Engineering, California, USA, March, 2013.

[11] M. I. Hossein, "Dynamic Scaling of a Web-Based Application in a Cloud Architecture," Master Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, Feb. 2014.

[12] L. Moore, K. Bean, and T. Ellahi, "A Coordinated Reactive and Predictive Approach to Cloud Elasticity," The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization, Valencia, Spain, 2013.

[13] D. Breitgand, E. Henis, and O. Shehory, "Automated and Adaptive Threshold Setting: Enabling Technology for Autonomy and Self-Management," the 2nd International Conference on Automatic Computing, Turin, Italy, 2005.

Archive of SID

www.SID.ir

Journal Of Electronical & Cyber Defence
Vol. 4, No. 4, 2017, Serial No. 16

A Runtime Auto-Scaling Mechanism for Multi-Tenant Software

A. Ghafori, M. Sharifi*

*Iran University of Science and Technology

(Received: 28/02/2015, Accepted: 03/05/2016)

ABSTRACT

Multi-Tenant software acts out based on auto-scale instance applications due to fluctuating workloads affected by increasing or decreasing the number of tenant demands and requested resource volumes. The aim of auto-scaling is better resource utilization of resource computing as well as provision of good Quality of Service (QoS) to tenants. Auto-scaling approach for meeting its used goal has challenges that consist of predicting workload, identifying correct resource requirement according to workload and selecting the policy of system for resource allocation with the lowest cost. In this research, a runtime auto-scaling mechanism is provided for multi-tenant software in order to answer mentioned challenges. First, due to lack of comprehensive description of a runtime system of multi-tenant software and based on found evidence of various studies, an architecture for runtime system of multi-tenant software is provided and a computing model is introduced that has three various granularity for its module of multi-tenant software. Then by providing an architecture for auto-scaling module of runtime system of multi-tenant software and using sub -modules that have appropriate functionality, mentioned challenges of auto-scaling are answered. To evaluate this, the proposed auto-scaling mechanism is implemented in the cloudism simulation environment and is tested and compared to the recent similar work in the same environmental conditions. The obtained results indicate that the provided auto-scaling mechanism compared to similar work despite the close number of violation of QoS, has better utilization and wastes computing resources less [13/28%-16/19%] with 99% confidence level.

Key words: Multi-Tenancy, Workload, Auto-Scaling, Resource Utilization, Quality of Service.

Archive of SID

* Corresponding Author Email: msharifi@iust.ac.ir

www.SID.ir