

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

نرم افزارهای چندمستاجر به دلیل نوسان بارکاری که متاثر از افزایش یا کاهش تعداد درخواست مستاجرها و حجم منبع درخواستی آنها است، نسبت به مقیاس خودکار نمونه های نرم افزاری اقدام می نمایند. هدف از مقیاس خودکار، بهره‌وری بهتر از منابع محاسباتی همراه با ارائه کیفیت سرویس مطلوب به مستاجرها است. رویکرد مقیاس خودکار در راه برآورده سازی هدف به کار گرفته شده خود با چالش‌هایی روبه‌رو است که این چالش‌ها متشکل از پیش‌بینی بارکاری، شناسایی کردن نیاز منابع درست مطابق بارکاری و چگونگی سیاست اتخاذی سیستم به منظور تخصیص منبع با کمترین هزینه است. در این مقاله با ارائه یک مکانیزم مقیاس خودکار زمان اجرا برای نرم افزارهای چندمستاجر راهکاری برای پاسخگویی به چالش‌های ذکر شده ارائه شده است. در ابتدا به دلیل عدم وجود تصویری جامع از سیستم زمان اجرا یک نرم افزار چندمستاجر، با توجه به شواهد یافت شده حاصل از بررسی‌ها و مطالعات گوناگون، به ارائه یک معماری برای سیستم زمان اجرا نرم افزار چندمستاجر پرداخته شده و مدل رایانش با سه دانه‌بندی برای مولفه‌ی نرم افزار چندمستاجر آن معرفی می‌گردد. سپس در ادامه با ارائه یک معماری برای مولفه مقیاس خودکار سیستم زمان اجرا نرم افزارهای چندمستاجر و به‌کارگیری زیرمولفه‌هایی با کارکردهای مناسب، جهت پاسخگویی به چالش‌های ذکر شده برای مقیاس خودکار اقدام می‌شود. به‌منظور ارزیابی، مکانیزم مقیاس خودکار ارائه شده در محیط شبیه‌ساز کلاسیک پیاده‌سازی شده و با جدیدترین کار مشابه در شرایط کاملاً یکسان مورد آزمایش و مقایسه قرار می‌گیرد.

واژه‌های کلیدی: چندمستاجری، بارکاری، مقیاس خودکار، بهره‌وری منبع، کیفیت سرویس

۱- مقدمه

به‌خصوص زمان پاسخ اشاره کرد. امروزه راه‌های مختلفی برای رفع این چالش‌ها انجام شده است که یکی از برجسته‌ترین آنها به‌کارگیری یک مکانیزم مقیاس خودکار 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 نرم افزاری گردیده و موجب دلسردی مشتریان و تاثیر منفی درآمد خواهد شد. از سوی دیگر، اگر ظرفیت برنامه‌ریزی شده متناسب با بارکاری اوج باشد منابع محاسباتی در بیشتر اوقات بیکار باقی خواهند ماند. مقیاس خودکار رویکردی است که به‌منظور غلبه بر این چالشها در محیط محاسباتی مورد استفاده قرار می‌گیرد [۲].

فراهم‌ساز خودکار منابع مجازی شده ارائه شده در زیرساخت محیط محاسبات ابری، امکان پیاده‌سازی برنامه‌های کاربردی با افزایش و کاهش مقدار منابع استفاده شده در این محیط را فراهم می‌آورد. این قابلیت مقیاس خودکار نامیده می‌شود و هدف اصلی آن تعدیل خودکار مقیاس سامانه‌ای است که برای برآورده کردن بارکاری متفاوت با حداکثر بهره‌وری منبع و باکیفیت سرویس

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 گذاشته شده 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، مورد توجه سازمان‌های فناوری اطلاعات قرار گرفته است. استفاده از نرم‌افزارهای چندمستاجر علی‌رغم مزایایی همچون مقرون‌به‌صرفگی، چالش‌هایی را نیز با خود به دنبال دارد که مقیاس‌پذیری از جمله 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

مؤلفه‌های این مکانیزم و کارکرد هر یک 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. در نهایت 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^۱ 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 مقیاس خودکار فراهم می‌کند. فن‌های مقیاس خودکار 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 مستاجر یا کاربران را فراهم می‌آورد [۴]. همه راه‌حل‌های مقیاس خودکار ارائه‌شده به کاربران خود این اجازه را می‌دهند که با در نظر گرفتن 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 کیفیت سرویس 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 مقیاس خودکار 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 مقیاس خودکار 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، رویکرد مقیاس و کیفیت سرویس است.

معین، برنامه‌های کاربردی را راه‌اندازی می‌کند [۳]. رویکرد مقیاس خودکار نیز در راه برآورده‌سازی هدف اصلی خود با چالش‌هایی روبه‌رو است که این چالش‌ها در ادامه ذکر می‌شوند. اولین چالش که یک رویکرد مقیاس خودکار پیش‌بینی بارکاری است [۲]. رویکرد مقیاس خودکار در محیط‌های محاسباتی مختلف به‌عنوان مثال یک محیط ابری دربردارنده فراهم‌سازی و آزادکردن منابع مطابق تغییرات بارکاری نرم‌افزار چندمستاجر در طی زمان است. آزادکردن منابع آسان است در صورتی که به-دست‌آوردن منابع، نرم‌افزار چندمستاجر را متحمل سربار کارکردی می‌نماید. دومین چالش شناسایی کردن نیاز منابع درست، مطابق با بارکاری ورودی است. از آنجایی که تعداد مستاجرها در هر وقفه زمانی متغیر است، تعداد درخواست‌های منابع لازم نیز متفاوت است. تعداد منابع لازم تابعی از تعداد مستاجرها، ماهیت نمونه برنامه کاربردی و همچنین نوع فراخوانی‌هایی است که هر مستاجر روی نمونه برنامه کاربردی 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 مقیاس خودکار با در نظر گرفتن آن‌ها دست به مقیاس کردن می‌نمایند. از سویی دیگر معیارهای مختلفی از کیفیت سرویس در توافقنامه سطح سرویس میان مستاجرها و ارائه‌دهندگان سرویس ذکر شده و ارائه‌دهندگان سرویس موظف به تضمین آن می‌شوند. مکانیزم مقیاس خودکار ارائه‌شده در این پژوهش در بعد اندازه دست به مقیاس زده و تعداد نمونه‌های برنامه کاربردی و ماشین مجازی را افزایش یا کاهش می‌دهد و همچنین با مدنظر قراردادن معیارهای کارکردی کیفیت سرویس، جهت بهره‌وری بهتر از منابع محاسباتی با حداقل نقض زمان پاسخ 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 مؤلفه‌های حاضر در آن را معرفی می‌نماید و در فصل چهارم، مؤلفه مکانیزم مقیاس خودکار به صورت خاص مورد بررسی قرار گرفته و زیر

۱-۱-۲- رویکردهای مقیاس

فن‌ها و راه‌حل‌های مقیاس خودکار 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 مقیاس خودکار با توجه به وضعیت جاری سیستم شروع به تصمیم‌گیری و اقدام می‌نماید. اغلب راه‌حل‌هایی مقیاس خودکار 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 مقیاس خودکار از فن‌های پیچیده برای پیش‌بینی تقاضای آینده استفاده کرده و به آماده‌سازی منابع مطابق با پیش‌بینی می‌پردازند. پیش‌بینی از آن جهت حائز اهمیت است که همواره یک تاخیر زمانی از زمان اجرای اقدامات مقیاس خودکار تا تاثیر آن وجود دارد که این زمان جهت آماده‌سازی و راه‌اندازی منابع مورد نظر است [۶-۷].

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

۲-۱-۲- نوع کیفیت سرویس

فن‌ها یا راه‌حل‌های مقیاس خودکار می‌بایست همچنین به‌وسیله‌ی نگه‌داری از یک کیفیت سرویس قابل قبول، از درستی کارکرد برنامه‌کاربردی حمایت نمایند. کیفیت سرویس به دو نوع توافق‌نامه سطح سرویس وابسته است: ۱- توافق‌نامه سطح سرویس برنامه‌کاربردی 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 زمان پاسخ، بهره‌وری از منبع، توان عملیاتی و طول صف 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 قرار گرفته‌اند.

۱-۲-۲- کارهای مرتبط دانشگاهی

منظور از کارهای مرتبط دانشگاهی محصولات ارائه‌شده در مقالات و پایان‌نامه‌های علمی است که با ارائه الگوریتم یا محصولی راه‌حلی را به منظور مقیاس خودکار 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 زمان پاسخ، به مقیاس خودکار منابع زیرساخت پرداخته و جهت بهبود بهره‌وری منابع اقدام می‌نماید. به دلیل واکنشی بودن این الگوریتم مقیاس خودکار هر بار به محض نقض زمان پاسخ انجام‌شده و این به کرات منجر به نقض توافق‌نامه سطح سرویس و تحمیل هزینه جریمه حاصل از آن می‌شود. از سویی دیگر مقیاس خودکار توسط این الگوریتم نیازمند برخی ملاحظات است که وجود آن‌ها همیشه امکان‌پذیر نیست. به‌عنوان مثال به منظور انجام مقیاس خوددرمانی می‌بایست ماشین مجازی با منبع بیکار وجود داشته باشد و یا به منظور 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 ماشین مجازی تنها در صورت داشتن منابع کافی از سوی ارائه‌دهنده فیزیکی امکان‌پذیر می‌شود.

در کاری دیگر آقای کجراوال [۱۰] به ارائه فن‌های جدیدی به منظور 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. از این‌رو می‌توان میزان بهره‌وری از منابع را به‌عنوان معیار اصلی آن عنوان کرد. در الگوریتم‌های یادشده تعداد گره‌هایی که به ازای هر بار اقدام مقیاس خودکار افزوده یا کاسته می‌شود به‌عنوان ورودی ذکر شده و معیار معینی برای چگونگی تعیین تعداد گره‌ها برای این ورودی ذکر نشده است. از سویی دیگر به منظور انجام مقیاس خودکار 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 گاهی 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 به منظور تکامل سریع و توان‌بالا برای کسب‌وکار و همچنین به منظور ملاقات نیازمندی‌های برنامه‌های کاربردی وب‌محور ایشان پرداخته شده است. این اثر با هدف غلبه بر چالش‌های ماهیت غیرقابل

محاسباتی با کمترین نقض معیارهای کیفیت سرویس ارائه شده در توافق نامه سطح سرویس است

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

جدول (۱). دسته بندی ارائه شده برای کارهای مرتبط

نوع کیفیت سرویس	نوع رویکرد مقیاس	انواع کارهای مرتبط	
بهره‌وری از منبع و زمان پاسخ	پیشگیرانه	مکانیزم مقیاس بویا مبتنی بر الگوریتم کجراوال	کارهای مرتبط دانشگاهی
		فن‌هایی برای رد پای ابری در حال بهینه	
		مقیاس نمودن منابع سبک‌وزن برای برنامه‌های کاربردی ابری	
بهره‌وری از منبع	واکنشی	ابر محاسباتی کشسان آمازون	تجاری
		میکروسافت آزر	
		رایت‌اسکیل	
بهره‌وری از منبع	واکنشی	اکالیپتوس	من‌باز
		اسکیل آر	
		کلادیا	

۳- معماری سیستم زمان اجرا نرم افزار چندمستاجر

معماری سیستم زمان اجرا که در این پژوهش برای سیستم نرم‌افزار چندمستاجر ارائه شده است متشکل از سه مولفه اصلی با نام‌های مولفه توزیع بار، مولفه مقیاس خودکار و مولفه 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

پیش‌بینی ترافیک‌های تقاضای کاربران برنامه‌های کاربردی و در نتیجه منابع محاسباتی مورد تقاضای آن‌ها، یک مکانیزم مقیاس خودکار ارائه کرده است. در مکانیزم مقیاس خودکار ارائه شده، به منظور تعیین مقدار آستانه مناسب برای معیارهای کیفیت سرویس اتخاذی و عدم نقض توافق نامه سطح سرویس از الگوریتم پیشنهادی کجراوال استفاده شده است و تمامی موارد تحلیلی ذکر شده در بالا در مورد این اثر نیز صدق می‌کند.

۲-۲-۲ کارهای مرتبط صنعتی

مقصود از کارهای مرتبط صنعتی محصولات مقیاس خودکار ارائه شده توسط سازمان و شرکت‌های بزرگ ارائه‌دهنده سرویس‌های ابری و چندمستاجری است. کارهای مرتبط صنعتی که در زمینه مقیاس خودکار 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 ارائه شده توسط این شرکت‌ها تصویر درست و نسبتاً جامعی از اصول مقیاس خودکار 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 مقیاس خودکار می‌پردازند. در میان شرکت متن‌باز ابری نیز شرکت‌های اکالیپتوس، اسکیل آر و کلادیا به منظور ارائه سرویس مقیاس خودکار نسبت به توسعه و یا استفاده از مولفه‌های مقیاس خودکار اقدام کرده‌اند. نتیجه بررسی شرکت‌های یادشده به این حقیقت اشاره دارد که، کارهای مرتبط صنعتی ارائه شده از روش‌های قوانین محور واکنشی استفاده می‌کنند. در این روش‌ها شرایط مقیاس کردن روی یک معیار هدف (به صورت اختیاری و یا اجباری) و با آستانه‌ای مشخص تعیین می‌شود و به محض رسیدن به آستانه‌ی تعیین شده توسط کاربرها و یا شرکت‌های سرویس دهنده دست به مقیاس خودکار می‌زنند [۱۲]. فراتر از آستانه‌های ثابت تعیین شده [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 به منظور بهره‌وری بهتر از منابع

بهره‌گیری از فناوری مجازی‌سازی و با ارائه یک ماشین مجازی روی ناظر ماشین مجازی، محیطی با توان زیرساختی معین و متاثر از میزان منابع فیزیکی لایه زیرین، برای اجرای نمونه‌های نرم‌افزاری ارائه می‌دهد. از آنجایی که هر ماشین مجازی 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 در بردارنده نمونه‌های ماشین فیزیکی است. منظور از ماشین فیزیکی، 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 ماشین فیزیکی، چندین ماشین مجازی روی آن در حال اجرا است.



شکل (۲). مدل رایانش نرم‌افزار چندمستاجر

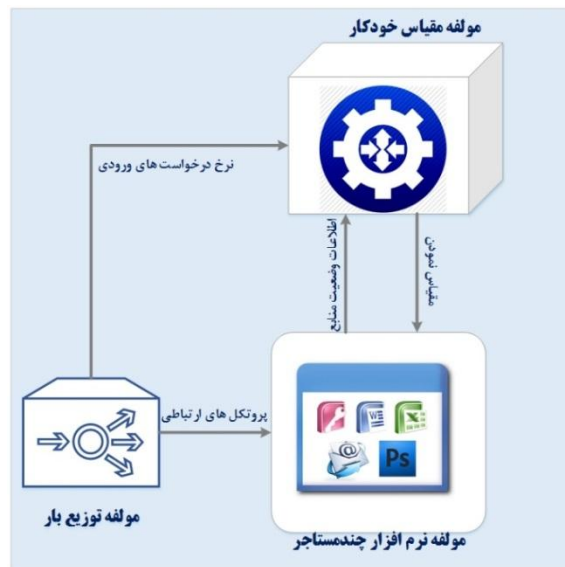
۳-۲- مولفه توزیع بار

مولفه توزیع بار ارائه شده در معماری سیستم زمان‌اجرا چندمستاجری، مولفه نرم‌افزاری است که ابتدا درگاهی را که کاربران یا مستاجرها به منظور دستیابی 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 توزیع بار 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

ادامه هر یک از مولفه‌های ذکر شده در معماری سیستم چندمستاجر معرفی و مولفه مقیاس خودکار به صورت کامل تشریح می‌گردد.



شکل (۱). معماری سیستم زمان‌اجرا نرم‌افزار چندمستاجر

۱-۳- مولفه نرم‌افزار چندمستاجر

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

۱-۱-۳- سطح‌های دانه‌بندی مدل رایانش نرم‌افزار

چندمستاجر

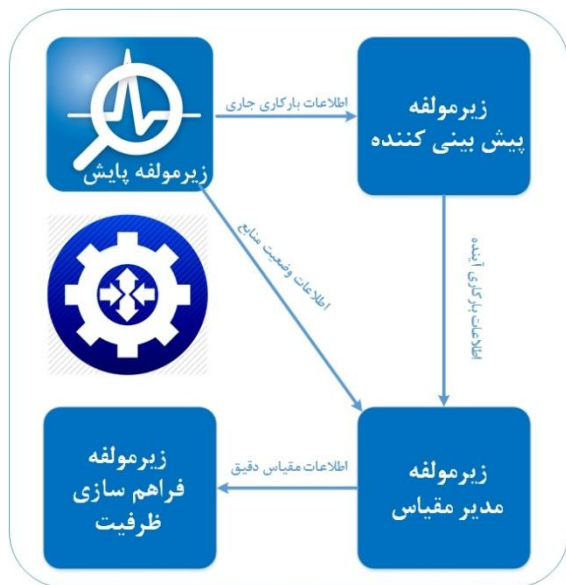
موجودیت‌های مدل رایانش معرفی شده برای نرم‌افزارهای چندمستاجر مطابق شکل (۲) خود متشکل از سه سطح دانه‌بندی است که این سطح‌ها به شرح زیر هستند.

۱-۱-۱-۳- نمونه برنامه کاربردی

سطح اول در بردارنده 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.

۲-۱-۱-۳- نمونه ماشین مجازی

سطح دوم در بردارنده نمونه‌های ماشین مجازی است که با



شکل (۳). معماری مولفه مقیاس خودکار

۴-۱- زیرمؤلفه پایش

این زیرمؤلفه وظیفه پایش و نظارت بر دو مولفه اصلی توزیع‌کننده بار و مولفه نرم‌افزار چندمستاجر را برعهده دارد. این زیرمؤلفه با پایش و نظارت بر نرخ ورودی تعداد درخواست مستاجرها به مولفه توزیع بار اطلاعات لازم در مورد بارکاری ورودی بر نرم‌افزار چندمستاجر را استخراج کرده و در اختیار زیرمؤلفه‌ی 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 زیرمؤلفه مدیرمقیاس ارائه می‌دهد.

زیرمؤلفه پایش اولین زیرمؤلفه مکانیزم مقیاس خودکار است که به ایفای نقش می‌پردازد. این مولفه در هر یک وقفه‌های زمانی که دارای اندازه‌های برابر بوده و با $t_{interrupt}$ نمایش داده می‌شود، مولفه توزیع‌کننده بار را پایش می‌نماید. در نتیجه این پایش اطلاعات متعددی در مورد بارکاری سیستم از جمله تعداد درخواست‌های سرویس ورودی مستاجرها و میزان متوسط مصرف برخی از منابع، از جمله 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 توان پردازشی (p) مصرفی به ازای چندین درخواست ورودی با زمان پاسخ معین (t)، در تعداد آزمایش معین اندازه‌گیری می‌شود. با توجه به برخی از اطلاعات بارکاری از جمله تعداد درخواست ورودی در هر وقفه زمانی ($t_{interrupt}$)، میزان منابع

زمان اجرا نرم‌افزارهای چندمستاجر، مکانیزم مقیاس خودکار است. هدف اصلی این مقاله ارائه یک مکانیزم مقیاس خودکار زمان اجرا برای نرم‌افزارهای چندمستاجر است که در بخش‌های قبل دلائل و ضرورت انجام آن بیان گردید.

مکانیزم مقیاس خودکار به منظور ایفای نقش درست و مقیاس نمودن 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 توزیع بار 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، ماشین مجازی و ماشین فیزیکی توسط مکانیزم مقیاس خودکار دریافت می‌شود و با توجه به اطلاعات رسیده از مولفه‌های توزیع‌کننده بار و نرم‌افزار چندمستاجر، عملیات مقیاس خودکار توسط مکانیزم مقیاس خودکار و روی نرم‌افزار چندمستاجر انجام می‌شود. مکانیزم مقیاس خودکار برحسب نیاز و با هدف حداکثر بهره‌وری از منابع موجود و همچنین حداقل اتلاف منابع، نسبت به مقیاس خودکار (افزایش یا کاهش نمونه‌ها) دانه‌بندی‌های مختلف در مدل رایانش نرم‌افزار چندمستاجر اقدام می‌نماید.

۴- معماری مکانیزم مقیاس خودکار

مکانیزم مقیاس خودکار که خود به‌عنوان یک مولفه در معماری سیستم زمان اجرا نرم‌افزارهای چندمستاجر به ایفای نقش می‌پردازد، دربردارنده‌ی چهار زیرمؤلفه است که هرکدام بخشی از فرآیند مقیاس خودکار را عهده‌دار هستند. در این بخش معماری مکانیزم مقیاس خودکار معرفی شده و سپس مولفه‌های تشکیل‌دهنده و تعاملات میان آن‌ها را به صورت کامل تشریح می‌شوند.

مولفه مکانیزم مقیاس خودکار متشکل از چهار زیرمؤلفه یا موجودیت است که عبارت‌اند از:

- ۱- زیرمؤلفه پایش
 - ۲- زیرمؤلفه پیش‌بینی‌کننده
 - ۳- زیرمؤلفه مدیر مقیاس
 - ۴- زیرمؤلفه فراهم‌سازی ظرفیت
- در شکل (۳) زیرمؤلفه‌های مربوط به مکانیزم مقیاس خودکار و تعاملات میان آن‌ها به نمایش داده شده است.

متحمل جریمه نیز می گرداند. راه حل مطلوب به منظور جلوگیری از سربراهای، به خصوص سربرای افزایش زمان پاسخ و همچنین به حداقل رساندن هزینه جریمه نقض کارکرد درج شده در توافق نامه سطح سرویس و هزینه فراهم سازی منابع، نیاز به تخصیص منابع در قالب تکثیر نمونه های برنامه کاربردی و ماشین مجازی پیش از فرارسیدن موعد اوج نیاز منابع است. این مهم تنها از طریق پیش بینی بارکاری آینده و به وسیله داده های تاریخی، امکان پذیر است. راه حل ما بارکاری سیستم را از ابتدا تا وقفه زمانی فعلی دریافت و با استفاده از آن بارکاری را در وقفه زمانی بعدی پیش بینی می کند. در این استراتژی ما بارکاری سیستم که شامل میزان تقاضای توان پردازشی توسط درخواست های ورودی است را در وقفه های زمانی مختلف ثبت و سپس با توجه به داده های تاریخی به دست آمده، به پیش بینی میزان بارکاری در وقفه زمانی بعد اقدام می کنیم. فن های مختلفی برای پیش بینی بارکاری در ریاضیات آماری وجود دارد و ما با بررسی و بر اساس دقت پیش بینی روش $ARIMA^1$ را انتخاب نمودیم که بر اساس شواهد به دست آمده یکی از دقیق ترین پیش بینی کننده ها برای سری های زمانی است و به دلیل نزدیکی مقدار پیش بینی شده با مقدار واقعی خطای پیش بینی بسیار کم و قابل قبولی ارائه می دهد.

۴-۲-۱- پیش بینی بارکاری ورودی با استفاده از $ARIMA$

فرض می کنیم سری زمانی $W_{t_{interrupt}}$ کل بارکاری ناشی از درخواست های ورودی باشد که میزان مصرف منبع r در وقفه زمانی $t_{interrupt}$ را نشان می دهد. در ادامه ما در معادلات زیر برای راحتی آن را با $W_{t_{interrupt}}$ نمایش می دهیم.

در پیش بینی یک قدم 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 $t_{interrupt}$ پیش بینی کنیم. اگر سری زمانی $W_{t_{interrupt}}$ از یک مدل $ARIMA(n, q)$ پیروی می کند و آن گاه در صورت ایستاد بودن و برای هر $t_{interrupt}$ معادله شماره یک مدل $ARIMA$ به شرح زیر است:

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

که در آن θ_j و θ_0 ثابت های تخمین زده شده از داده موجود هستند. اصطلاح ϵ_k خطاهایی را به نمایش می گذارد که به صورت مستقل، مشابه مثال های توزیع شده از یک توزیع نرمال با میانگین صفر و واریانس محدود σ^2 فرض شده اند. پارامترهای n و q به

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

جدول (۲). محاسبه میزان بارکاری به وسیله زیرمولفه پایت مکانیزم مقیاس خودکار

معرفی متغیرها:

$E_t(X) = X$ امید ریاضی مصرف منبع

μ حداکثر زمان پاسخ قابل قبول

N_R تعداد درخواست سرویس ورودی توسط مستاجر

W_X میزان منبع مصرفی توسط نوع منبع

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

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

قدم دوم: محاسبه مقادیر بارکاری W بر اساس میزان واحد

پردازش مرکزی مصرفی p در وقفه زمانی $t_{interrupt}$

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

۴-۲-۲- زیرمولفه پیش بینی کننده

این زیرمولفه اطلاعات استخراجی در مورد نرخ ورود درخواست مستاجر را از زیرمولفه پایت دریافت کرده و با بهره گیری از مدل های ریاضی به پیش بینی پرداخته و از تحلیل مدل های آماری به وسیله سری های زمانی استفاده می کند. این زیرمولفه از این طریق نسبت به پیش بینی بارکاری ورودی یا نرخ ورودی درخواست مستاجر در آینده، اقدام می نماید.

رویکرد مقیاس خودکار در نرم افزارهای چندمستاجر همواره درگیر فراهم سازی و آزاد کردن منابع در قالب نمونه های برنامه کاربردی و نمونه های ماشین مجازی است که میزان آن متناسب با تغییرات بارکاری یا به عبارتی تعداد تقاضای سرویس مستاجر در واحد زمان است. این فراهم سازی و آزاد کردن منابع می بایست به صورت صحیح انجام گیرد تا با فراهم سازی بیش از نیاز منابع موجب افزایش هزینه منابع مصرفی و یا با آزاد کردن ناصحیح و برنامه ریزی نشده منابع موجب کاهش کیفیت سرویس ارائه شده و نارضایتی مستاجر نگردد. از سویی دیگر به رغم آسان بودن آزاد کردن، فراهم سازی نمونه های برنامه کاربردی و نمونه های ماشین مجازی خود سربراهای عملکردی برای نرم افزار چندمستاجر به ارمان می آورد. یکی از این سربراهای عملکردی که در این پروژه به طور خاص مورد بررسی است سربرای افزایش زمان پاسخ است که افزایش آن از حد مجاز تعیین شده در توافق نامه سطح سرویس بین مستاجر و ارائه دهنده سرویس علاوه بر نارضایتی مشتری، سرویس دهنده نرم افزار چندمستاجر را

مولفه مدیر مقیاس به صورت شبه کد در جدول (۳) نمایش داده شده است.

جدول (۳). الگوریتم انجام عملیات مقیاس خودکار به وسیله مولفه مدیر مقیاس

ورودی:	
$u_{t_{interrupt}}^{AI}$: تغییرات تعداد نمونه برنامه کاربردی فعال در وقفه زمانی $t_{interrupt}$	
$u_{t_{interrupt}}^{VMI}$: تغییرات تعداد نمونه ماشین مجازی فعال در وقفه زمانی $t_{interrupt}$	
خروجی:	
$VMI_Adjustment$: متغیر شرطی اعلان کاهش مقیاس نمونه ماشین مجازی در وقفه زمانی بعدی	
$AI_Adjustment$: متغیر شرطی اعلان کاهش مقیاس نمونه برنامه کاربردی در وقفه زمانی بعدی	
<pre> 1 Begin: 2 If (VMI_Adjustment = true) 3 { Deprovision($u_{t_{interrupt}}^{VMI}$); VMI_Adjustment ← false ; } 4 If (AI_Adjustment = true) 5 { Deprovision($u_{t_{interrupt}}^{AI}$); AI_Adjustment ← false ; } 6 Switch $u_{t_{interrupt}}^{AI}$ 7 Case 1 ($u_{t_{interrupt}}^{AI} > 0$): 8 If ($u_{t_{interrupt}}^{VMI} > 0$) 9 { Provision ($u_{t_{interrupt}}^{VMI}$); 10 Provision ($u_{t_{interrupt}}^{AI}$); 11 } else if ($u_{t_{interrupt}}^{VMI} = 0$) 12 { Provision ($u_{t_{interrupt}}^{AI}$); 13 Break; 14 Case 2 ($u_{t_{interrupt}}^{AI} = 0$): 15 Break; 16 Case 3 ($u_{t_{interrupt}}^{AI} < 0$): 17 If ($u_{t_{interrupt}}^{VMI} < 0$) 18 { VMI_Adjustment ← true; 19 AI_Adjustment ← true; 20 } else { AI_Adjustment ← true } 21 Break; 22 Return VMI_Adjustment, AI_Adjustment; 23 End; </pre>	

پیش بینی عکس تا زمان وقوع دقیق و کاهش میزان تقاضا کاهش مقیاس صورت نمی گیرد. هدف مکانیزم ما کنترل تعداد نمونه های ماشین مجازی و برنامه کاربردی برای کاهش هزینه های سیستم به خصوص در هزینه مصرف منابع محاسباتی شامل توان پردازشی و همچنین تا حد امکان هزینه نقض توافق نامه سطح سرویس با ارائه سرویس با زمان پاسخ مناسب است. در همین راستا این مکانیزم مقیاس خودکار با تخصیص درست نمونه های ماشین مجازی و برنامه کاربردی با گذشت زمان و متناسب با تغییرات بار کاری و تعداد درخواست های سرویس موجب به حداقل رساندن تعداد نمونه های برنامه کاربردی و ماشین مجازی بیکار و در نتیجه حداقل سازی میزان مصرف منبع و هزینه های متاثر از آن می شود. از سویی دیگر، از آنجایی که مقیاس خودکار خود به دلیل افزودن و راه اندازی کردن تعدادی از نمونه های برنامه کاربردی و ماشین مجازی نیارمند اندکی زمان است مکانیزم مقیاس خودکار با فراهم سازی معین نمونه های برنامه کاربردی و ماشین مجازی پیش از موعد تقاضا موجب کاهش سر بار زمانی اضافی و کاهش زمان پاسخ به مستاجرها شده و هزینه ناشی از جریمه نقض توافق نامه سرویس را به حداقل ممکن می رساند.

همان طور که در بخش های قبل ذکر گردید، نحوه مدیریت مقیاس کردن تعداد نمونه ها در انواع مختلف ماشین مجازی و نمونه برنامه کاربردی بر عهده مولفه مدیر مقیاس و همچنین تصمیم گیری راجع به چگونگی و کیفیت فراهم سازی آن بر عهده مولفه فراهم سازی ظرفیت است که در ادامه به آن می پردازیم. بعد از این که میزان بار کاری برای زمان یا زمان های بعد از زمان جاری توسط مولفه پیش بینی کننده، پیش بینی و محاسبه شد این اطلاعات به منظور بررسی و تصمیم گیری به مولفه مدیر مقیاس داده می شود. این مولفه در وهله اول تعداد نمونه برنامه کاربردی و تعداد نمونه های لازم برای پاسخگویی به بار کاری را توسط تابع زیر محاسبه و با تعداد نمونه های برنامه کاربردی یا نمونه های ماشین مجازی جاری مقایسه می کند و تعداد نمونه های اضافی یا کم را به ازای بار کاری جاری محاسبه می کند.

$$f_1(W_{t_{interrupt}+1}^r) = x_{t_{interrupt}+1}^{AI} = \frac{W_{t_{interrupt}+1}^r}{C_{r,AI}} \quad (7)$$

$$f_2(W_{t_{interrupt}+1}^r) = x_{t_{interrupt}+1}^{VMI} = \frac{W_{t_{interrupt}+1}^r}{C_{r,VMI}} \quad (8)$$

حال با توجه به محاسبه $u_{t_{interrupt}}^I$ که نشانگر تعداد ماشین افزایش یا کاهش یافته میان دو وقفه کاری متوالی است فرمان مقیاس خودکار از سوی مدیر مقیاس صادر گردیده و جهت اعمال و انجام درست آن، اطلاعات لازم به مولفه فراهم سازی ظرفیت فرستاده می گردد. الگوریتم انجام عملیات مقیاس خودکار توسط

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 مقیاس خودکار بعد از انجام تصمیم‌گیری توسط مولفه مدیرمقیاس و با فراخوانی دو تابع Provision و Deprovision توسط مولفه فراهم‌سازی ظرفیت صورت می‌گیرد. در صورتی که $u_{interrupt}^I$ 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 در صورتی که $u_{interrupt}^I$ برابر با یک عدد مثبت باشد با فراخوانی تابع Provision، بر اساس نوع ورودی عملیات افزایش مقیاس با افزودن نمونه‌های برنامه‌کاربردی و ماشین‌مجازی در لایه‌های مربوطه انجام می‌شود.

در صورتی که $u_{interrupt}^I$ برابر با یک عدد مثبت باشد این بدان معنا است که تعداد نمونه‌های نوع I می‌بایست افزایش یابد؛ بنابراین فرمان افزایش مقیاس 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 شبیه‌سازی 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 مقیاس خودکار پیشنهادی را در شبیه‌ساز کلاسیک پیاده‌سازی نمودیم. برای اثبات تحقق یافتن اهداف ذکرشده، مکانیزم مقیاس خودکار پیشنهادی را با مکانیزم مقیاس خودکار 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 داده‌های یکسان مورد آزمایش قرارداده و نتایج حاصل را ارزیابی نمودیم.

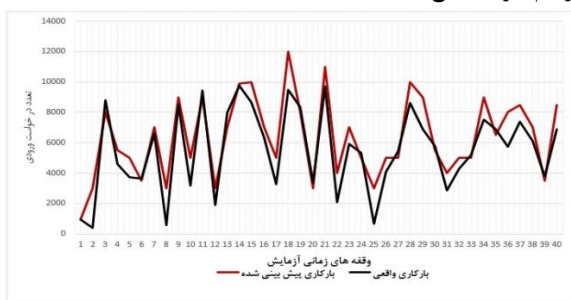
نرم‌افزارهای چندمستاجر با توجه به نوع ارائه سرویس معمولاً دارای تعداد درخواست سرویس‌های نوسانی و بارکاری متفاوت در طی زمان هستند. به همین جهت برای آزمودن و ارزیابی ۱۰

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 مقیاس خودکار پیشنهادی

شکل (۵) نیز شمایی از آزمایش اول انجام روی کار مشابه مقیاس خودکار پیشنهادی شده است 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،

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 می‌توان نتیجه گرفت که مکانیزم پیشنهادی با ۹۹ درصد اطمینان دارای (۱۶/۱۹-۱۳/۲۸) درصد بهره‌وری در استفاده از توان پردازشی و همچنین دارای اتلاف کمتر توان پردازشی نسبت به کار مشابه انجام شده است.

یکی از اهداف دیگر مکانیزم مقیاس خودکار پیشنهادی به حداقل رساندن تعداد دفعات نقض کیفیت سرویس در عین افزایش بهره‌وری و کاهش اتلاف منابع محاسباتی است. در همین راستا و به منظور ارزیابی از این جنبه نیز، تعداد دفعات نقض زمان پاسخ در مکانیزم مقیاس خودکار 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 زمان پاسخ در مکانیزم پیشنهادی و کار مشابه

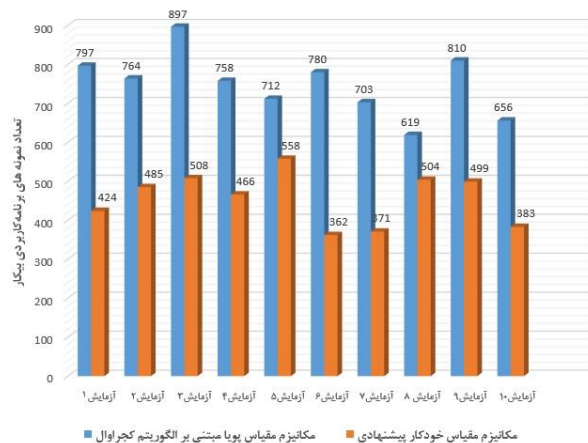
همان‌طور که از نتایج به نمایش درآمده در شکل (۷) پیداست، تعداد دفعات نقض زمان پاسخ در مکانیزم مقیاس خودکار پیشنهادی نسبت به مکانیزم مقیاس پویا مبتنی بر الگوریتم کجراوال مقدار اندکی رشد داشته است. تعداد حداقل و حداکثر دفعات نقض زمان پاسخ در مکانیزم پیشنهادی به‌طور متوسط ۵ بار بیشتر از مکانیزم دیگر است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

درخواست‌های کاربران در نهایت توسط نمونه‌های برنامه‌کاربردی دریافت و سرویس‌دهی می‌شوند.

به‌علاوه از آنجایی که پاسخگویی به بارکاری جاری می‌بایست با هدف به حداقل رساندن دفعات نقض کیفیت سرویس باشد تعداد دفعات نقض زمان پاسخ نیز محاسبه می‌گردد. در اینجا علی‌رغم وجود انواع مختلفی از منابع سامانه‌ای تنها منابع پردازشی یعنی واحد پردازش مرکزی و در میان انواع ویژگی‌های کیفیت سرویس تنها کارکرد و به صورت بسیار دقیق زمان پاسخ 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 شبیه‌سازی و ارزیابی

به منظور مقایسه، نتایج حاصل از آزمایش‌های متعدد و گوناگون روی مکانیزم مقیاس خودکار پیشنهادی و کار مشابه، در یک نمودار ستونی کنار هم قرار داده شده است. در شکل (۶) تعداد نمونه‌های برنامه‌کاربردی بیکار و بدون استفاده در طی آزمایش‌های متعدد برای مکانیزم مقیاس خودکار و کار مشابه نمایش داده شده 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 توان پردازشی در مکانیزم مقیاس خودکار 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 مقیاس خودکار پیشنهادی با کارمشابه اقدام

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. Calvacchia, B. A. Capraescu, 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. Lorigo-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.

پیش‌بینی شده 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، در صورتی که میزان منابع پیش‌بینی شده کمتر از میزان منابع لازم برای بارکاری ورودی باشد، در آن صورت به دلیل تخصیص بارکاری بیشتر از ظرفیت به منابع موجود زمان پاسخ متناسب با حجم بارکاری مازاد تحت تاثیر قرار گرفته و بیشتر می‌شود. همان‌طور که پیداست وقوع حالت دوم موجب رشد اندک دفعات نقض زمان پاسخ در مکانیزم پیشنهادی نسبت به کار مشابه شده است.

۶- نتیجه‌گیری

ما در این مقاله با ارائه یک مکانیزم مقیاس خودکار 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 مولفه مقیاس خودکار سیستم زمان‌اجرا نرم‌افزارهای چندمستاجر و به‌کارگیری زیرمولفه‌هایی با کارکردهای مناسب، در جهت پاسخگویی به چالش‌های مقیاس خودکار در نرم‌افزارهای چندمستاجر اقدام نمودیم.

حوزه مورد بحث در این مقاله به سبب وسعت و تازگی دارای چالش‌های متعددی است. موضوعاتی همچون چندمستاجری و مقیاس خودکار هر کدام به‌تنهایی به دلیل نو بودن و همچنین پیچیدگی دارای چالش‌ها و موارد قابل بررسی مختلفی هستند. سعی بر آن 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 مقیاس خودکار 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 معیار کیفیت سرویس مورد پایش توسط مکانیزم مقیاس خودکار به‌منظور میزان نقض کیفیت سرویس تنها زمان پاسخ بوده است که می‌تواند برای سایر معیارهای کیفیت سرویس همچون توان خروجی، دسترسی‌پذیری بررسی و تحلیل گردد.

A Runtime Auto-Scaling Mechanism for Multi-Tenant Software

A. Ghafari, 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.