زمان دریافت مقاله: ۹۵/۴/۲۶
زمان پذیرش مقاله: ۹۵/۷/۲۹

مقیاس‌بندی خودکار منبع ابری: رویکردهای نوین در گزینش VM مازاد

محمد صادق اصلان‌پور
دانشجوی کارشناسی ارشد گروه مهندسی کامپیوتر، واحد سیرجان، دانشگاه آزاد اسلامی، سیرجان، ایران
پست الکترونیکی: aslanpour.sadegh@gmail.com

سید ابراهیم دشتی رحمت‌آبادی*
استادیار گروه مهندسی کامپیوتر، واحد جهرم، دانشگاه آزاد اسلامی، جهرم، ایران
پست الکترونیکی: sayed.dashty@gmail.com

چکیده

ارائه‌دهندگان سرویس کاربرد[1] (ASP) با هدف کاهش هزینه‌های مالی به اجاره ماشین مجازی[2] (VM) ارائه شده در محیط ابری روی می‌آورند. مهمترین مشخصه رایانش ابری که می‌تواند به ASP در کاهش هزینه کمک کند، خاصیت مقیاس‌پذیری[3] منابع ابری است. از این رو ASP با تدارک سازوکاری جهت مقیاس‌بندی خودکار[4] منابع اجاره شده، سعی می‌کند از تامین بیش از نیاز[5] و تامین کمتر از نیاز[6] جلوگیری کند. محققان به ارائه سازوکارها و بررسی پارامترهای موثر در تصمیم‌گیری این سازوکارها می‌پردازند. اما بجز نحوه دستیابی به یک تصمیم مقیاس‌بندی، چگونگی اجرا کردن تصمیم از اهمیت بالایی برخوردار است. از دید نویسندگان این مقاله، این‌که در هنگام اجرای تصمیم مقیاس‌بندی پایین[7] کدام ماشین اجاره شده برای آزادسازی انتخاب شود یک چالش اساسی است که می‌تواند در عملکرد سازوکار تاثیرگذار باشد. در این مقاله با شبیه‌سازی چرخه کامل مدیریت خودکار منابع در شبیه‌ساز کلادسیم[8]، رویکردهای نوینی برای گزینش VM مازاد ارائه می‌شوند. تفاوت چشمگیر در نتایج و عملکرد سازوکار مقیاس‌بندی در اثر اعمال هر کدام از رویکردهای پیشنهادی، بر اثربخشی و اهمیت سیاست‌گذاری در گزینش VM مازاد دلالت می‌کنند. همچنین نتایج نشان داد که رویکردهای آگاه به بار (از لحاظ کمی و کیفی) می‌توانند به بهبود کیفیت سرویس (۱۸ ٪) و رویکرد آگاه به هزینه[9] به کاهش هزینه (۲۰ ٪) کمک کنند.

واژه‌های کلیدی: رایانش ابری، مقیاس‌بندی خودکار، کیفیت سرویس، ماشین مجازی

* نویسندهٔ مسئول

1- Application Service Providers - ASP
2- Virtual Machine - VM
3- Scalability
4- Auto-Scaling
5- Over-Provisioning
6- Under-Provisioning
7- Scale Down or Scale In
8- CloudSim
9- Cost-Aware

۱ مقدمه

با گسترش خدمات ارائه شده در محیط ابری، امروزه بسیاری از ASPها تمایل دارند که زیرساخت‌های ارائه شده توسط ارائه‌دهندگان سرویس ابر[10] (CSP) را برای میزبانی کاربرد خود برگزینند. آمازون با ارائه سرویس EC2[11] به‌عنوان یکی از معروف‌ترین CSPها در بازار کسب وکار اینترنتی شناخته می‌شود. یکی از مهمترین مزیت‌های مهاجرت از زیرساخت سنتی به ابری در قابلیت مقیاس‌بندس خودکار منابع ابری است [۱]. چون ASP به پویایی فضای وب و درخواست‌های کاربران نهایی به کاربرد واقف است و می‌داند که نرخ ورود درخواست کاربران به کاربرد همیشه به یک اندازه نیست [۲, ۳]. با این پیش‌زمینه، یک ASP در کنار اجاره منابع ابری (در قالب VM) و سرویس‌دهی به درخواست‌های کاربران، به یک سازوکار مقیاس‌بندی خودکار برای مدیریت VMها نیاز دارد [۱]. هدف این سازوکار جلوگیری از به هدر رفتن هزینه اجاره منابع (تامین بیشتر از نیاز) و بهبود سطح رضایت کاربران (تامین کمتر از نیاز) است.

مقیاس‌بندی منابع مجموعه‌ای از فرایندها است که به طور مداوم تکرار می‌شوند؛ این فرایندها به طور کلی به چهار مرحلهٔ پایش[12]، تحلیل[13]، برنامه‌ریزی[14] و اجرا[15] تقسیم می‌شوند (MAPE)[16] [۴, ۵]. مرحلهٔ پایش تعیین می‌کند که سرکشی به وضعیت منابع در چه زمانی (یا زمان‌هایی) باید صورت بگیرد. پس از فراخوانی پایشگر و جمع‌آوری اطلاعات به تحلیل آن‌ها پرداخته می‌شود. عموما تحلیل وضعیت منابع [۶–۱۰] یا توافق‌نامهٔ سطح سرویس (SLA)[17] [۴, ۱۱, ۱۲] مورد توجه قرار می‌گیرد. در تحلیل وضعیت SLA به موقعیت پارامترهای مرتبط با کیفیت سرویس[18] (QoS) مثل زمان پاسخی که کاربران تجربه می‌کنند یا تعداد درخواست‌های کاربران که منجر به نقض SLA[19] شده پرداخته می‌شود. پس از دریافت تحلیل‌های موثر، نوبت به نتیجه‌گیری از تحلیل‌های به‌دست آمده یا به عبارتی برنامه‌ریزی جهت اتخاذ تصمیم مقیاس‌بندی می‌رسد. تصمیم می‌تواند مقیاس‌بندی بالا[20] یا پایین[21] منابع باشد. اجرای دستور مقیاس‌بندی آخرین مرحله از چرخهٔ مدیریت منبع است. در این مرحله همه آنچه که پایش، تحلیل و برنامه‌ریزی شده با ارسال دستوری به ارائه‌دهنده سرویس ابر به ثمر می‌نشیند. چالش مورد بررسی در این تحقیق به این مرحله برمی‌گردد. چرا که در اجرای تصمیم مقیاس‌بندی پایین به دلیل وجود تعداد زیادی VM اجاره شده، مطمئنا اجرا کنندهٔ تصمیم با چندین VM نامزد روبرو است که باید از بین آن‌ها یکی را به عنوان مازاد انتخاب و آزادسازی کند. اینجا دو سوال بی پاسخ مانده، این که آیا اهمیتی دارد که کدام VM انتخاب شود؟ و اگر جواب مثبت است چه رویکردی بهتر است برای این گزینش در نظر گرفته شود؟ برای پاسخ به این سوالات نیاز است که رویکردهایی در این مرحله قرار داده شوند و اثربخشی آن‌ها بر روی عملکرد سازوکار مقیاس‌بندی مورد ارزیابی قرار گیرند. در این تحقیق با ارائه ۵ رویکرد مبتکرانه در گزینش VM مازاد به سوالات مطرح شده پرداخته می‌شود. ما در کار قبلی [۱۳] با ارائه یک سازوکار مقیاس‌بندی خودکار بر روی مرحله برنامه‌ریزی از چرخه MAPE متمرکز شدیم؛ در حالی که در این تحقیق تمرکز بر روی مرحلهٔ اجرا است. به علاوه در آن تحقیق برای اولین بار به طور مقدماتی به مبحث گزینش VM مازاد با ارائه رویکرد آخرین بالا اولین پایین[22] (LUFD) پرداخته شد. سهم علمی این تحقیق عبارت است از:

– ارائه رویکردهای خلاقانه در گزینش VM مازاد در محیط ابری

– بررسی اثربخشی رویکردهای متنوع در گزینش VM مازاد

10- Cloud Service Providers - CSP
11- Elastic Compute Cloud – EC2
12- Monitor
13- Analyze
14- Plan
15- Execute
16- Monitor – Analyze – Plan - Execute
17- Service Level Agreement - SLA
18- Quality of Service - QoS
19- SLA Violation
20- Scaling Up
21- Scaling Down
22- Last Up First Down - LUFD

– شبیه‌سازی رویکردهای پیشنهادی در کلادسیم

در ادامـــه این مقاله، بخش ۲ مروری بر کارهای مرتبط را نشان می‌دهد. رویکردهای پیشنهادی برای گزینش VM مازاد در بخش ۳ آمده؛ به علاوه در بخش ۴ یک سازوکار مقیاس‌بندی بـــرای کاربرد ابری ارائه می‌شـــود که از این رویکردهـــا در تصمیم‌های مقیاس‌بندی پایین بهره می‌برد. بخش ۵ به شبیه‌سازی سازوکار مورد نظر در کلادسیم و ارزیابی نتایج حاصله می‌پردازد. در نهایت بخش ۶ حاوی نتیجه‌گیری و پیشنهادها است.

۱- کارهای مرتبط

بحث در خصوص مقیاس‌بندی خودکار منبع در محیط ابـــری از دیدگاه‌هـــای مختلف مورد توجه قـــرار می‌گیرید. در [۶] روشـــی پیش‌فعال به کمک شبکه عصبی مصنوعی ارائه شد که ســـازوکار آن‌ها قابلیت مقیاس‌بندی بر اساس کســـب آگاهی از بهره‌وری منابع را داشـــت. آن‌ها عملکرد شـــبکه‌های عصبی را با یکدیگر مقایسه کردند ولی از SLA غافل شـــدند. نویسندگان [۷] روشی پیش‌فعال آگاه به منابع را برای ارائه‌دهنده سرویس کاربرد در ابر ارائه دادند. آن‌ها سازوکار خود را در کلادسیم پیاده کردند ولی در ارزیابی‌ها به مســـئلهٔ هزینه که از مهمترین نیازهای ASP است توجه نشد. در [۱۴] روشی پیش‌فعال آگاه به منابع و SLA ارائه شد ولی در ارزیابی نتایج تنها به معیارهای مرتبط با سری‌های زمانی پرداختند و از مبحث هزینه و کیفیت ســـرویس غافل شدند. آن‌ها در کار بعدی [۱۵] سعی کردند سازوکار خود را در بارکاری شدیدتری ارزیابی کنند.

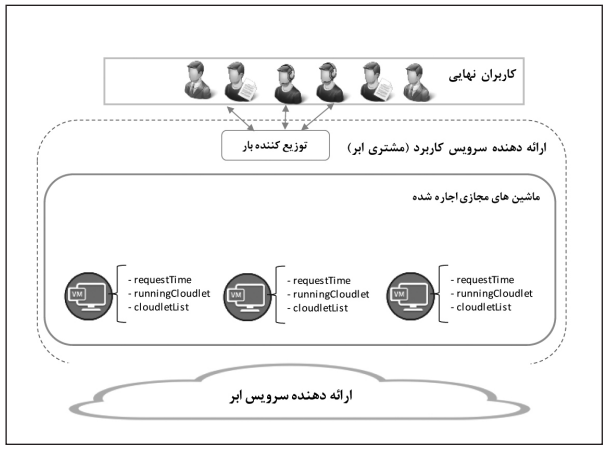
نویسندگان [۴] به مسئلهٔ مقیاس‌بندی منبع آگاه به SLA به شیوه واکنشی مبتنی بر چرخه MAPE پرداختند. آن‌ها از تجزیه و تحلیل معیارهای مرتبط با بهره‌وری منابع، هزینه و نقض SLA صرف نظر کردند. در [۱۱] یک معماری برای مدیریت منابع ارائه شد که مجهز به سازوکار مقیاس‌بندی افقی واکنشـــی است. روش پیشـــنهادی از آگاهی SLA در قالب زمان پاسخ برخوردار است. آن‌ها در ارزیابی عملکرد ســـازوکار خود به مســـئلهٔ هزینه بدون مد نظر قرار دادن جریمـــه نقـــض SLA پرداختنـــد. در [۸] الگوریتمـــی برای مقیاس‌بندی خودکار منابع با روشی پیش‌فعال ارائه شد که برتریش نســـبت به تامین منبع ایستا نشان داده شد. روش آن‌ها برای تصمیم‌گیـــری، از وضعیت بهره‌وری منابع در تنظیم‌کننده بار بهره می‌برد. هزینه بدون محاســـبه جریمه نقض SLA مورد بررسی قرار گرفت.

در کاری دیگر [۹] مســـئلهٔ مقیاس‌بندی خودکار با ارائه چارچوبی مطرح شد که تنها از وضعیت منابع آگاه است و SLA را مورد نظر قرار ندادند. در این کار تحلیل تنظیم‌کننده بار تنها راهنما در تصمیمات مقیاس‌بندی است. هرچند در ارزیابی‌ها بـــه معیارهای مرتبط با SLA و هزینه پرداختند اما جریمـــه نقض SLA را در این محاســـبات تاثیر ندادند. نویســـندگان [۱۲] تکنیک‌های آگاه به QoS را برای مدیریت منابـــع پیشـــنهاد دادند. با توجه به گســـتردگی کار، اما در شبیه‌ســـازی‌ها از در نظر گرفتن تاخیر در راه‌اندازی VM غافل شـــدند. نقطه قوت کار آن‌هـــا در تامین منابع ناهمگن بود اما از آن طرف به مســـئلهٔ نقض SLA و جریمه‌های آن نپرداختند. در تحقیقی دیگر [۱۰] یک معماری برای مدیریت منابع ارائه شده که به کمک روشی واکنشی آگاه به منابع به مقیاس‌بندی منابع می‌پردازد. تحلیل‌هایی که این سازوکار بـــرای تصمیم‌گیری در اختیار دارد تنها از بهره‌وری منابع (حافظـــه) آگاهی دارد و بـــا ســـازوکارهای آگاه به SLA مقایســـه نشـــد. آن‌ها همچنین از ارزیابی سازوکار نسبت به معیارهای مرتبط با کیفیت ســـرویس و هزینه صرفنظر کردند.

از این رو روشـــن اســـت که مســـئلهٔ چگونگی گزینش VM مـــازاد در زمان اتخاذ تصمیمـــات مقیاس‌بندی پایین از دیـــد محققان پنهان مانده اســـت. به علاوه کارآمدی یک ســـازوکار با ارزیابی جامع معیارها روشن می‌شود که ما به معیارهای زیر توجه داریم:

– کیفیت سرویس در قالب زمان پاسخ

– نوسان در بهره‌وری منابع



شکل ۱: نمای کلی از موقعیت و مشخصات منابع اجاره شده توسط ASP از CSP

- سربار مقیاس‌بندی در اثر تصمیمات مقیاس‌بندی
- هزینه تحمیلی به ASP

۲- رویکردهای پیشنهادی برای گزینش VM مازاد

در ایـــن بخش پنج رویکرد خلاقانـــه برای گزینش VM مـــازاد در زمان اتخاذ تصمیمات مقیاس‌بندی پایین معرفی می‌شـــوند. شـــکل ۱: نمای کلی از موقعیت و مشـــخصات منابـــع اجـــاره شـــده توســـط □□□ از □□□ نمـــای کلی از مشخصه‌هایی را نشان می‌دهد که می‌توانند یک VM اجاره شـــده را از دیگری متمایز سازند. این ویژگی‌ها عبارتند از requestTime، runningCloudlet و cloudletList. بـــه ایـــن معنی که ســـازوکار مقیاس‌بندی در عملیات اجرای دستور مقیاس‌بنـــدی پایین بـــه منظور گزینش VM مـــازاد، با این ویژگی‌هـــا روبرو اســـت و از آن‌هـــا می‌توانـــد در اعمال سیاســـت‌های مختلف بهره بگیرد. ویژگـــی requestTime نشان‌دهنده زمان شروع اجاره VM از CP که درگزینش‌های وابسته به زمان و وابســـته به هزینه می‌تواند موثر باشد. چـــرا که معمولا هزینه و صدور صورتحســـاب برای VM اجاره شـــده به مدت زمان ســـپری شـــده از requestTime مرتبط اســـت. ویژگی runningCloudlet نشان‌دهنده تعداد درخواســـت کاربران است که برای پردازش به VM محول شده‌اند. این ویژگی به سیاســـت‌های گزینش VM مازادی

الگوریتم ۱ رویکرد آخرین بالا اولین پایین

```
1   Input:      On-Demand Vm List
2   Output:     surplusVm
3   Param-      lastRequestTime ← MIN_VAL-
    eters:      UE
4   for vm in onDemandVmList do
5       If vm.getRequestTime() > lastRequestTime
        then
6           lastRequestTime ←
            vm.getRequestTime();
7           surplusVm ← vm;
8       end if
9   end for
10  return surplusVm;
```

مرتبط می‌شود که نگاهی کمّی به بار هر VM دارد. همچنین ویژگی cloudletList بیانگر مقدار حجم بار در حال پردازش توســـط VM اســـت که اثر این ویژگی می‌تواند به گزینش با رویکردهای آگاه به بار منابع مرتبط شـــود. در ادامه به معرفی چهـــار رویکرد خلاقانه که از ایـــن ویژگی‌ها بهره می‌برند پرداخته می‌شود.

۲-۱- رویکرد آخرین بالا اولین پایین

این رویکرد، گزینش آخرین VM که تامین شـــده را به عنوان VM مازاد در دســـتور کار خود قرار می‌دهد. از این رو طبق الگوریتم ۱ کل VM از نوع بر حســـب تقاضا[23] را بررســـی و ســـپس از بین آن‌ها به دنبال VM با بزرگترین زمان درخواست (بر اساس ویژگی requestTime) می‌گردد. منظور زمان درخواست VM از CSP است.

۲-۲ - رویکرد اولین بالا آخرین پایین (FUFD)

در این رویکرد همیشـــه قدیمی‌ترین VM از نظر زمان درخواســـت انتخاب می‌شود (بر اســـاس ویژگی request-Time). لـــذا طبـــق الگوریتـــم ۲ رویکرد اولین بـــالا اولین پایینالگوریتم ۲ در این روش پس از دریافت لیســـت VM، جدیدترین VM انتخاب می‌شود.

23- On-demand

الگوریتم ۲ رویکرد اولین بالا اولین پایین

```
1  Input:    On-Demand Vm List
2  Output:   surplusVm
3  Param-    firstRequestTime ← MAX_VALUE
   eters:
4  for vm in onDemandVmList do
5      if vm.getRequestTime() < firstRequestTime
       then
6          firstRequestTime ←
           vm.getRequestTime();
7          surplusVm ← vm;
8      end if
9  end for
10 return surplusVm;
```

۲-۳ - رویکرد آگاه به کلادلت (Cloudlet-Aware)

از آنجا که با آزادسازی یک VM ممکن است یک یا چند کلادلت در حین اجرا در آن VM متوقف شوند و اجرای مجدد آن‌ها توسط VM دیگر موجب نقض SLA شود، در نتیجه روشی که این مسئله را در گزینش VM مازاد لحاظ کند می‌تواند در کاهش نقض SLA موثر باشد. از این رو این رویکرد طبق الگوریتم ۳ با استعلام از زمانبند هر VM از تعداد کلادلت در حال اجرا توسط VM مطلع می‌شود و VM که تعداد کلادلت کمتری برای اجرا داشته باشد را به عنوان مازاد انتخاب می‌کند (بر اساس ویژگی runningCloudlet).

۲-٤ رویکرد آگاه به بار (Load-Aware)

در روش آگاه به کلادلت مشخص شد که می‌توان VM که تعداد کلادلت کمتری را در اختیار دارد به عنوان مازاد انتخاب کرد، اما از آنجا که کلادلت‌ها (که نمایانگر درخواست کاربران به وب هستند) از حجم و نیاز پردازشی یکسانی برخوردار نیستند، لذا ممکن است یک VM تعداد کلادلت کمتری در اختیار داشته باشد اما مدت زمان بیشتری برای پردازش نیاز باشد. دلیل این تنوع در نیاز پردازشی کلادلت‌ها از معماری سه لایه کاربرد

الگوریتم ۳ رویکرد آگاه به کلادلت

```
1  Input:    On-Demand Vm List
2  Output:   surplusVm
3  Param-    minRunningCloudlets ← MAX_VAL-
   eters:    UE
4  for vm in onDemandVmList do
5      runningCloudlets ← vm.getScheduler().run-
       ningCloudlets();
6      if runningCloudlets < minRunningCloudlets
       then
7          minRunningCloudlets ← runningCloud-
           lets;
8          surplusVm ← vm;
9      end if
10 end for
11 return surplusVm;
```

وب سرچشمه می‌گیرد. زیرا درخواست‌های مربوط به هر لایه نیاز پردازشی متفاوتی را طلب می‌کند [۱۶]. در الگوریتم ارائه شده برای این روش ابتدا سعی می‌شود که میزان حجم باقیمانده از پردازش برای هر VM با استعلام از زمان‌بند همان VM محاسبه شود (بر اساس ویژگی cloudletList) و آنگاه VM که کمترین بار را داشته باشد به عنوان مازاد انتخاب می‌شود (الگوریتم ۴ را ببینید). پردازش باقیمانده از هر کلادلت موجود در یک VM بر اساس برآورد زمان سپری شده از پذیرش کلادلت و نیاز پردازشی آن نسبت به توان پردازشی VM محاسبه می‌شود. از آنجا که در سیستم مورد نظر ما یک کلادلت با نیاز پردازشی X هسته‌ای نمی‌تواند بر روی یک VM با توان پردازشی X+۱ هسته‌ای، زودتر از موعد مشخص اجرا شود، لذا این این مسئله هم توسط الگوریتم مد نظر قرار گرفته (خط ۱۲ تا ۱۴) است.

۲-۵- رویکرد آگاه به هزینه (Cost-Aware)

در گزینش VM مازاد وقتی که بحث بر سر ارائه‌دهندگان

الگوریتم ۴ رویکرد آگاه به بار

```
1  Input:          On-Demand Vm List
2  Output:         surplusVm
3  Parameters:     minLoad ←MAX_VALUE
4  cloudletList ←new ArrayList <Cloudlet> ()
5  for vm in onDemandVmList do
6      vmRemainedLoad ←0;
7      cloudletList ←vm.getScheduler().getCloudletList();
8      for cloudlet in cloudletList do
9          length ←cloudlet.getLength() * cloudlet.getNumberOfPes();
10         ranTime ←clock - cloudlet.getFirstSubmissionTime();
11         remainedLength ←length - (ranTime * (vm.getMips() * vm.getNumberOfPes() );
12         if vm.getNumberOfPes() > cloudlet.getNumberOfPes() then
13             remainedLength ←length - (ranTime * ( vm.getMips() * cloudlet.getNumberOfPes() ) );
14         end if
15         vmRemainedLoad ← vmRemainedLoad + remainedLength;
16     end for
17     if vmRemainedLoad < minLoad then
18         minLoad ←vmRemainedLoad;
19         surplusVm ←vm;
20     end if
21 end for
22 return surplusVm;
```

سرویس ابر مثل آمازون است، یک نکتهٔ مهم و اساسی در جهت کاهش هزینه برای ASP وجود دارد و آن این‌که هر CSP یک حدنصاب برای صدور صورتحساب تعیین می‌کند.

الگوریتم ۵ رویکرد آگاه به هزینه

```
1  Input:          On-Demand Vm List
2  Output:         surplusVm
3  Parameters:     maxPassedTimeFromLastHour ←MIN_VALUE
4  for vm in onDemandVmList do
5      passedTimeFromLastHour ←null;
6      availableTime ←clock - vm.getRequestTime();
7      passedTimeFromLastHour ←availabletime % anHour; // % = mod
8      if passedTimeFromLastHour = 0 then
9          passedTimeFromLastHour ←anHour;
10     end if
11     if passedTimeFromLastHour > maxPassedTimeFromLastHour then
12         maxPassedTimeFromLastHour ←passedTimeFromLastHour;
13         surplusVm ←vm;
14     end if
15 end for
16 rerutn surplusVm;
```

به‌عنوان مثال، آمازون در سرویس EC2 صورتحساب یک ساعته صادر می‌کند، لذا عمل مقیاس‌بندی پایین یک VM که مثلا ۳ ساعت و ۴۰ دقیقه از زمان اجاره آن سپری شده، صورتحساب ۴ ساعته را برای ASP در بر خواهد داشت. در نتیجه این نکته می‌تواند پیش‌زمینه‌ای برای تدارک یک روش گزینش VM مازاد آگاه به هزینه باشد (الگوریتم ۵ را ببینید). به نحوی که از بین گزینه‌های موجود، یک VM انتخاب شود که بیشترین دقایق از آخرین ساعت مصرف را سپری کرده یا به اصطلاح بیشترین بهره از آخرین ساعت خود را داشته است (بر اساس ویژگی re-questTime). در نتیجه انتظار می‌رود این رویکرد بتواند به کاهش هزینه کمک کند. الگوریتم پیشنهادی برای این رویکرد سعی می‌کند برای هر VM مدت زمان سپری شده از آخرین ساعت را (با عملگر mod) به‌دست آورد. VM که

بیشـــترین زمان را از آخرین ساعت مصرف خود سپری کـــرده به عنوان مازاد انتخاب می‌شـــود. یک VM که دقیقا در سر ساعت قرار دارد در بیشترین اولویت انتخاب است (مثلا دقیقا ۳ ساعت مصرف).

۳- سازوکار مقیاس‌بندی خودکار

همان‌طور که گفته شـــد، ســـازوکار مقیاس‌بندی جهت مدیریت خـــودکار منابع باید یک چرخـــه مدیریتی را طور مـــداوم تکرار کند. در این چرخه چهـــار عمل اصلی پایش، تحلیل، برنامه‌ریزی و اجرا به ترتیب طی می‌شوند. الگوریتم ۶ جزئیات ســـازوکار مورد نظر ما در طی این مراحل را به وضوح نشان می‌دهد. نکته: از آنجا که در تعیین سازوکار مقیاس‌بنـــدی منابع بجـــز نیازهای ASP بایـــد به نیازهای کاربران نهایی هم توجه کرد؛ لذا تصمیم‌گیری این سازوکار بر اساس وضعیت قرارداد SLA انجام می‌پذیرد (سازوکار آگاه به SLA). در سازوکار پیشنهادی، چرخه مدیریت منابع با احضار خودکار پایشـــگر هر از x دقیقه شروع می‌شود (مرحلهٔ پایش). در هر مرحله پایش ابتدا تحلیل‌های به دست آمـــده در طول یـــک دقیقه اخیر اندازه‌گیری شـــده و زمان پاســـخی را که کاربران تجربه کرده‌اند به‌دســـت می‌آورد. زمان پاسخ هر درخواست طبق فرمول۱ محاسبه می‌شود

فرمول ۱

ResponseTime=FinishTime-ProcessTime- ArrivalTime

به‌طوری که FinishTime زمان تکمیل درخواست، ProcessTime مدت زمان مورد نیاز برای اجرای درخواســـت توســـط VM (کـــه حاصل تقســـیم تعـــداد دســـتورالعمل درخواست وارده بر توان پردازشی VM است) و ArrivalTime زمان پذیرش درخواست توسط ASP است. میانگین زمان پاسخ درخواســـت‌ها به عنوان پارامتر تصمیم‌گیری به برنامه‌ریز ارسال می‌شود (مرحلهٔ تحلیل) تا با توجه به آستانه‌های تعریف شده برای زمان پاسخ در قرارداد SLA،

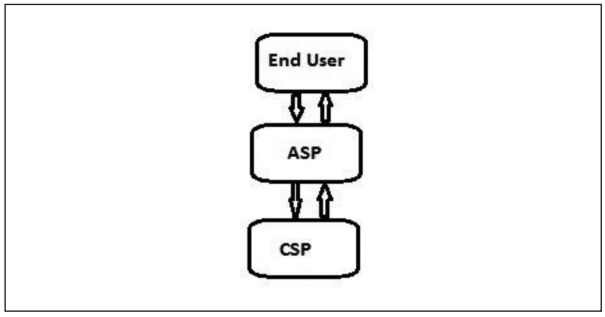
الگوریتم ۶ سازوکار مقیاسبندی خودکار

```
1   Data:        On-Demand VM List
2                SLAHistory
3                surplusVMSelectionPolicy
    Param-
4   eters:       x ← monitoringInterval;
5                avgResponsetime ← null;
66               decision ← doNothing;
7                surplusVM ← null;
8   repeat every x minutes
9     avgResponseTime ← Analyzer(SLAHistory);
10    /* Decision Making */
11    decision ← Planner (avgResponseTime);
12    if decision = SCALE_UP then
13      Execute (ScaleUp (request new VM));
14    else if decision = SCALE_ DOWN then
15      surplusVM ← surplusVMSelectionAp-
        proach (On-Demand VM List);
16      Execute (Scale Down (surplusVM));
17    end if
17  end
```

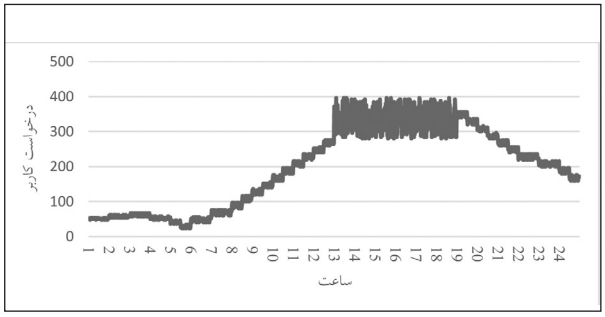
تصمیم مقیاس‌بندی اتخاذ شـــود (مرحلهٔ برنامه‌ریزی). در صورتی کـــه تصمیم مقیاس‌بندی بالا مـــد نظر برنامه‌ریز باشـــد، اجراکننده اقدام به درخواست یک VM جدید (خط ۱۳)؛ اما در صورتی که تصمیم بر مقیاس‌بندی پایین باشد (خط ۱۴ تا ۱۷)؛ آن زمان است که هنر سازوکار پیشنهادی نمایان می‌شـــود (مرحلهٔ اجرا). در اینجا ابتدا توســـط یکی از رویکردهای پیشنهادی برای گزینش VM مازاد اقدام به تشخیص مناسب‌ترین VM از بین VMهای برحسب تقاضا می‌شـــود و ســـپس آن VM را برای آزادســـازی احضار می‌کند. surplusVMSelectionApproach ماشین مازاد را به کمک یکی از رویکردهای معرفی شـــده در بخش ۲ انتخاب می‌کند.

۴- شبیه‌سازی

شبیه‌ساز کلادســـیم [۱۷] به دلیل پر کاربرد بودن در



شکل ۲: بازیگران اصلی ارائه و دریافت سرویس کاربرد در ابر



شکل ۳: درخواست کاربران به کاربرد وب در طول یک روز

زمینه‌های مختلف مدیریت منابع ابری [۷–۹, ۱۸–۲۰]، جهت شبیه‌سازی سازوکار مقیاس‌بندی انتخاب شد. کلادسیم یک بستهٔ جاوا مبتنی بر رویداد[۲۴] است که همه تغییرات در اجرای برنامه به واسطه رد و بدل کردن رویدادها صورت می‌گیرند . بازیگران اصلی ارائه و دریافت سرویس کاربرد در محیط ابر عبارتند از: کاربر نهایی، ASP و CSP؛ لذا باید هر کدام به عنوان یک موجودیت در کلادسیم در نظر گرفته شوند (شکل ۲ را ببینید).

موجودیت کاربر نهایی باید درخواست‌های کاربران را تولید و در فواصل زمانی تصادفی، شبیه به رفتار کاربران یک وبگاه واقعی (در قالب یک رویداد) به سمت ASP ارسال کند؛ پس موجودیت End-User با این ویژگی به کلادسیم اضافه شد. راه‌حل شبیه‌سازی درخواست‌های کاربران در کلادسیم، ترفند تاخیر در رویداد ارسالی به ASP است. تعداد درخواست‌ها و زمان ارسال آن‌ها باید از قبل مشخص شود. از این رو بارکاری وارده به کاربرد برای یک روز منطبق بر الگوی محیط واقعی ابر تولید شد. با توجه به دسته‌بندی سری‌های زمانی [۲۱]، این بار کاری در ساعات ۱۲ تا ۱۸ الگوی نامنظم و در سطح شبانه روز الگوی فصلی را نشان می‌دهد که به طور روزانه قابل تکرار است (شکل ۳ را ببینید). برای هماهنگ شدن سازوکار با بار وارده، از نوسان بار در ۴ ساعت ابتدایی کاسته شد. بار تولید شده جمعا شامل ۲۸۷۶۲۵ درخواست کاربر، به نحوی که حداقل ۲۴ و حداکثر ۳۹۶ درخواست در دقیقه توسط کاربرد وب به ASP منتقل می‌شوند. نیاز پردازشی درخواست‌ها ۲۵۰۰، ۵۰۰۰ و ۷۵۰۰ میلیون دستورالعمل به ترتیب برای لایه ۲۵ کاربرد، پایگاه داده و وب است.

پس از موجودیت کاربر نهایی، پیاده‌سازی ASP مستلزم توسعه موجودیت Broker در کلادسیم است. موجودیت ASP در این سیستم ۲ عدد ماشین ذخیره شده برای نیازهای دائمی و در ادامه چندین ماشین از نوع برحسب تقاضا برای پوشش نیازهای موقت آتی تامین می‌کند. تامین یا عدم تامین VM در حین اجرا توسط سازوکار مقیاس‌بندی پیشنهادی انجام می‌شود. تاخیر راه‌اندازی یک VM معادل ۱۰ دقیقه و آزادسازی به دلیل ناچیز بودن [۲۲] صفر در نظر گرفته شد. با توجه به اهمیت توجه به تاخیر در راه‌اندازی یک VM، پایشگر هر از ۱۰ دقیقه احضار می‌شود. از آنجا که ۵ رویکرد گزینش VM مازاد معرفی شد، لذا آزمایش با هر کدام از این رویکردها به طور مستقل انجام می‌شود. همان طور که گفته شد، پارامتر موثر در تصمیم‌گیری مقیاس‌بندی را زمان پاسخ در نظر گرفتیم (روش آگاه به SLA) از این رو آستانه زمان پاسخ برای مقیاس‌بندی بالا و پایین به ترتیب ۱ و ۰,۲ تعیین شد. از طرفی نویسندگان در [۴] زمان پاسخ بین ۷۰۰ تا ۱۲۰۰ میلی ثانیه و در [۱۸] زمان پاسخ ۱۰۰۰ میلی ثانیه را به عنوان معیار در نظر گرفتند. موجودیت ASP در کنار سازوکار مقیاس‌بندی به یک سازوکار مدیریتی برای بار وارده هم نیاز دارد، از این رو بار وارده (هر کلادلت) با سیاست گردش نوبت (Round Robin – RR) بین VMها توزیع می‌شوند.

24- Event-driven

25- Tier

از طرفی پیکربندی VMها با الگو گرفتن از سرویس EC2 آمازون صورت پذیرفت. جزئیات بیشتر در مورد پیکربندی VM در [۱۳] آمده است. زمانبندی کارها در VMs از نوع اشتراک زمانی[۲۶] تعیین شد. در خصوص موجودیت CSP هم باید گفت که به طور پیش‌فرض در کلادسیم وجود دارد. زیرا چند موجودیت مرکزداده می‌توانند نقش یک CSP را بازی کنند. به منظور استقرار لایه CSP، ۵ مرکز داده که هر کدام حاوی ۱۰ میزبان است راه‌اندازی شد.

٤-١- معیارهای ارزیابی

معیارهای کمّی و کیفی مهم در این مسئله آن‌هایی هستند که به SLA و هزینه به ترتیب برای کاربر نهایی و ASP مرتبط می‌شوند. از این رو معیارهای مورد بررسی عبارتند از:

- زمان پاسخ: یک سازوکار عادلانه باید زمان پاسخ (یا تاخیر) را در محدوده مشخص شده در قرارداد SLA حفظ کند. معیار حداقل ۰٫۲ و حداکثر ۱ ثانیه برای کاربردهای ابری با بررسی‌هایی به دست آمد [۱۳]. میانگین زمان پاسخ در انتهای اجرا به کمک میانگین گرفتن از کل زمان پاسخ‌های ضبط شده در هر دقیقه توسط تحلیلگر رفتار SLA محاسبه می‌شود (نحوه محاسبه زمان پاسخ درخواست‌ها در هر دقیقه در فرمول۱ مشخص شد). همچنین انحراف معیار (SD) زمان پاسخ را اندازه‌گیری می‌کنیم. مقدار پایین‌تر برای میانگین نشان از رضایت بیشتر کاربران و SD پایین‌تر نشان از تصمیمات بموقع و جلوگیری از تاخیرهای شدید در پاسخ‌دهی دارد.

- نقض SLA: درصدی از درخواست‌ها که زمان پاسخ بیشتر از ۱۰۰۰ میلی ثانیه را تجربه کرده‌اند؛ درصد بالا منجر به تحمیل جریمهٔ بیشتری به ASP می‌شود.

- هزینه: کارآمدی یک سازوکار مقیاس‌بندی در گرو توانایی در کاهش هزینه است. از آنجا که یک سازوکار ممکن است هزینه را کاهش دهد ولی آنقدر SLA را نقض کند که این کاهش هزینه به چشم نیاید، پس هزینه مورد نظر ما مجموع هزینه اجاره VMها به اضافه هزینه تحمیل شده در اثر جریمه‌های نقض SLA است (فرمول ۲ را ببینید).

فرمول ۲

Cost=RentingCost+SLA Penalty

به‌طوری‌که Cost کل هزینه (به دلار) تحمیل شده به ASP است. RentingCost هزینه اجاره کردن VMها که حاصل ضرب تعداد ساعات استفاده و هزینه یک ساعت اجاره هر VM است. به علاوه SLA Penalty به کمک فرمول۳ به‌دست می‌آید:

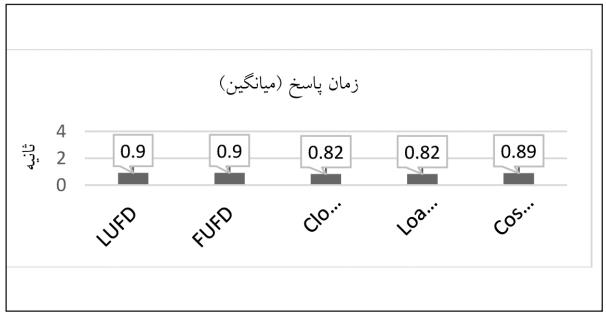
فرمول۳

$$SLA\ Penalty = Ceil\left(\frac{\sum_{i=1}^{SLAViolents}(ResponseTime_i - 1)}{60 * 60}\right) * Price$$

به‌طوری‌که SLA Penalty هزینه تحمیل شده به ASP در اثر نقض SLA، SLAViolents برابر است با درخواست‌هایی که زمان پاسخ بیشتر از ۱ ثانیه را تجربه کرده‌اند. لذا ابتدا مجموع زمان‌های (ثانیه‌ها) که تاخیر در اجرای درخواست‌ها در طول شبیه‌سازی رخ داده (با محاسبه $ResponseTime_i$-1 برای هر درخواست) به‌دست می‌آید؛ سپس مجموع ثانیه‌های تاخیر به ساعت تبدیل می‌شود (با تقسیم بر ۶۰ * ۶۰). عدد به‌دست آمده به کمک تابع Ceil به بالا گرد می‌شود، چون آمازون دقایق باقیمانده را هم به‌عنوان یک ساعت محاسبه می‌کند. در نهایت حاصل در Price (هزینه یک ساعت اجاره آن VM از نوع برحسب تقاضا) ضرب می‌شود تا جریمه نقض SLA به‌دست آید. در صورت تامین منبع ناهمگن، محاسبه قیمت به عامل تاخیرساز یعنی نوع VM بستگی دارد. با تاثیر دادن جریمه SLA در هزینه، دیگر با اطمینان می‌توان گفت که هزینه کمتر نشان از موفقیت یک سازوکار دارد. جهت بررسی عملکرد سازوکار، فقط هزینه ماشین‌های برحسب تقاضا محاسبه می‌شود، نه ماشین‌های ذخیره شده.

- سطح بهره‌وری منابع: ظرفیتی از منابع که در طول اجرا در حال استفاده بوده، تحت عنوان میانگین بهره‌وری منابع شناخته می‌شود. سطح بالاتر نشان‌دهنده تامین منبع

26- Time-Shared



شکل ۴: نتایج زمان پاسخ پس از اعمال رویکردهای مختلف گزینش VM مازاد

به اندازه و منطقی‌تر است. همچنین SD بهره‌وری منابع هم اندازه‌گیری می‌شود؛ مقدار پایین‌تر SD نشان‌دهنده غاقلگیر نشـــدن VMها از بار وارده است. جزئیات بیشتر در مورد نحوه محاسبه بهره‌وری در [۱۳] آمده است.

– تعـــداد VMهـــای تامیـــن شـــده (تعـــداد تصمیمات مقیاس‌بندی بالا): دفعاتی که سازوکار مقیاس‌بندی در طول شبیه‌سازی تصمیم به تامین یک VM می‌گیرد. تصمیمات کمتر نشان‌دهنده آگاهی به نوسانات، سربار کم و پرهیز از تصمیمات عجولانه است.

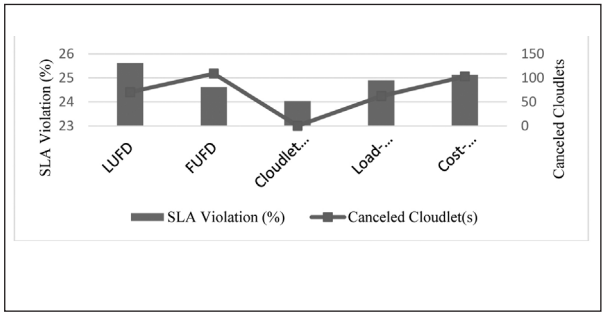
– حداکثر VM تامین شـــده: حداکثر VM تامین شـــده در اوج شـــدت بار را اندازه‌گیـــری می‌کند. تعداد VM کمتر نشان‌دهنده مقاومت سازوکار در برابر شرایط با بار شدید است.

٤-۲- نتایج

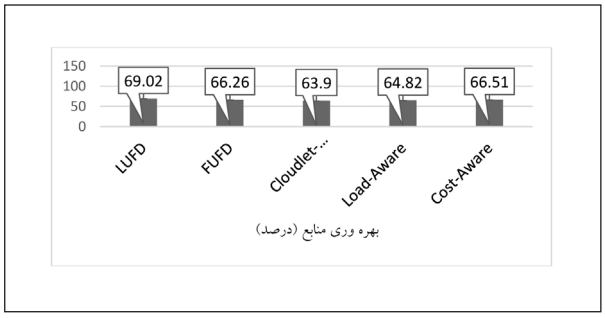
اینجا بـــه ارزیابی نتایج به‌دســـت آمـــده در اثر اعمال هـــر کدام از رویکردهای گزینش VM مازاد در ســـازوکار مقیاس‌بندی پرداخته می‌شود.

– زمان پاسخ، رویکرد گزینش VM مازاد آگاه به کلادلت و آگاه به بار بهترین عملکرد را داشتند. رویکردهای FUFD و LUFD ضعیف‌تر از بقیه در تاثیرگذاری بر زمان پاســـخ عمل کردند و به میزان ۱۸٪ تفاوت در نتایج مشـــاهده شد (شکل ۴ را ببینید).

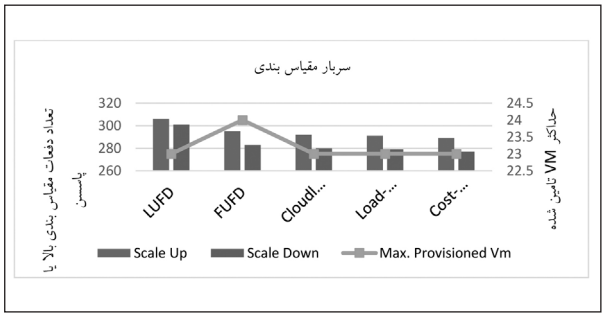
– نقض SLA، رویکرد آگاه به کلادلت توانســـت درصد نقـــض SLA را در بهترین حالت قـــرار داده و منجر به لغو



شکل ۵: نتایج SLA کاربران پس از اعمال رویکردهای مختلف گزینش VM مازاد



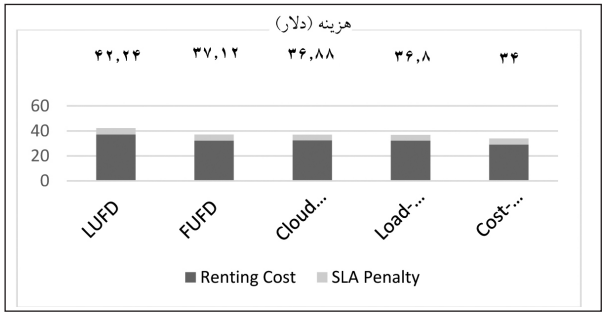
شکل ۶: نتایج بهره‌وری منابع پس از اعمال رویکردهای مختلف گزینش VM مازاد



شکل ۷: نتایج سربار مقیاس‌بندی پس از اعمال رویکردهای مختلف گزینش VM مازاد

تعداد کلادلت کمتری شـــود؛ به نحوی که رویکرد آگاه به کلادلت هیچ لغو درخواســـتی را در پی نداشـــت. نتایج به دست آمده یک رابطه نسبی بین درصد نقض SLA و تعداد درخواست‌های لغو شده را نشان می‌دهد (شکل ۵ را ببینید).

– بهـــره‌وری منابـــع، نتایـــج بهره‌وری منابـــع پس از اعمـــال رویکردهای مختلف برعکس نتایج به دســـت آمده در معیارهای زمان پاســـخ و SLA است. از نظر بهره‌وری منابـــع، رویکردهایی این معیار را در ســـطح بالاتری قرار دادنـــد که زمان پاســـخ را با تنزل بیشـــتری روبرو کردند



شکل ۸: نتایج هزینه پس از اعمال رویکردهای مختلف گزینش VM مازاد

(شکل ۶ را ببینید).

– سربار مقیاس‌بندی، تاثیر رویکردهای مختلف گزینش VM مازاد بر نتایج این معیار بسیار موثر است به‌نحوی که از نظر تعداد VM تامین شده (مقیاس‌بندی بالا) در بهترین حالت نسبت به بدترین حالت تفاوتی معادل ۶٪ دیده شد (شکل ۷ را ببینید).

– هزینه، رویکرد آگاه به هزینه طبق پیش‌بینی‌ها بهترین عملکرد را در کاهش هزینه نشان داد (۲۰ ٪ بهبود). روش‌های LUFD و FUFD بدترین تاثیر را بر هزینه داشتند. همچنین روش‌های آگاه به کلادلت و آگاه به بار هزینه را در حد متعادل‌تری قرار دادند (شکل ۸ را ببینید).

٤-۳- بحث

اینجا با توجه به نتایج به‌دست آمده به بحث در خصوص عملکرد مورد انتظار و رسالت هر رویکرد می‌پردازیم. رویکرد LUFD گزینش بر اساس آخرین VM درخواست شده را دنبال می‌کرد. به دلیل تعداد تصمیمات مقیاس‌بندی بالا در محیط عملیاتی، بالطبع در این رویکرد دفعات زیادی پیش می‌آید که یک VM که اخیرا (به تازگی) درخواست شده، به عنوان مازاد انتخاب شود. از این رو، نتایج ضعیف برای زمان پاسخ، درصد نقض SLA و هزینه بر این مسئله تایید کرد. سپس رویکرد FUFD ارائه شد تا برعکس روش قبل، گزینش VMهای قدیمی‌تر را در اولویت خود قرار دهد. این رویکرد به دلیل گزینش VMهای قدیمی‌تر که احتمال زیادتری داشت که به خوبی از آن‌ها استفاده شده (از نظر مدت زمان استفاده)، توانست از جنبه هزینه بهتر از رویکرد FUFD عمل کند.

بر خلاف دو رویکرد اول، رویکردهای دیگر از یک پارامتر به عنوان راهنما در گزینش VM مازاد بهره می‌برند. رویکردهای آگاه به کلادلت و آگاه به بار همان طور که انتظار می‌رفت با نگاهی به ترتیب کمیتی و کیفیتی به بار در حال اجرا در VM توانستد تاثیرات مثبتی بر زمان پاسخ، وضعیت SLA و در کل نیازهای کاربران نهایی بگذارند. در نهایت رویکرد آگاه به هزینه به خوبی به وظیفه خود عمل کرد و کم هزینه‌ترین گزینش‌ها را در تصمیمات خود نشان داد. در مجموع همه رویکردهای پیشنهادی به رسالتی که داشتند عمل کردند. اما تنها معیار بهره‌وری منابع بود که نتوانست نتایج معنی‌داری را به نمایش بگذارد.

۵- نتیجه‌گیری

در این تحقیق به بهبود تامین منبع خودکار منابع ابری پرداخته شد. یک سازوکار مقیاس‌بندی خودکار برای تامین منبع ارائه شد که اساس کار آن بر پایه چرخه MAPE مدیریت منابع است. آنچه که کار ما را با تحقیقات در این زمینه متمایز ساخت توسعه مرحلهٔ اجرا از چرخه MAPE است، جایی که تصمیم مقیاس‌بندی باید اجرا شود. لذا سه مرحلهٔ اول از چرخه MAPE در همه آزمایش‌ها یکسان در نظر گرفته شدند؛ همچنین در مرحلهٔ چهارم نحوه اتخاذ تصمیم مقیاس‌بندی بالا هم مشابه بودند. این نحوه اتخاذ تصمیم مقیاس‌بندی پایین بود که با اضافه شدن قابلیت گزینش VM مازاد دستخوش تغییرات شد. ما برای اجرای تصمیم مقیاس‌بندی پایین یک ابزار جالب معرفی کردیم که سازوکار را قادر خواهد ساخت که به صورت هدفمند اقدام به گزینش یک VM مازاد کند. از این رو پنج رویکرد خلاقانه ارائه شد و به منظور ارزیابی عملکرد این رویکردها از شبیه‌ساز کلادسیم بهره گرفتیم. تفاوت در نتایج به دست آمده، اهمیت و تاثیرگذاری اعمال رویکرد مختلف در گزینش VM مازاد را نمایان می‌کند. همچنین نتایج نشان داد که

رویکرد گزینش VM مازاد جهت اثربخشـــی بیشتر باید از حداقل از یک پارامتر آگاه باشـــد؛ رویکرد آگاه به پارامتر هزینه می‌تواند به نفع ASP در کاهش هزینه اثرگذاری خود را نشان دهد (تا ۲۰٪ بهبود) و رویکردهای آگاه به پارامتر بار (چه از نظر کمی و چه کیفی) بر روی معیارهای مرتبط بـــا QoS تاثیر مثبـــت می‌گذارند (تا ۱۸ ٪ بهبـــود). محققان می‌توانند با ترکیب آن‌ها سیاست‌های خاص خود را اعمال کنند؛ به‌عنوان مثال، ترکیـــب رویکردهای آگاه به هزینه و آگاه به بار برای موازنه بین هزینه و QoS. در کار آتی قصد داریم تصمیم‌گیری مقیاس‌بندی را تقویت کرده و در مرحلهٔ اجرای تصمیمات مقیاس‌بندی از رویکردی چندجانبه برای گزینش VM مازاد بهره ببریم.

منابع

[1] 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, vol. 12, pp. 559-592, 2014.

[2] M. Kihl, P. Ödling, C. Lagerstedt, and A. Aurelius, «Traffic analysis and characterization of Internet user behavior,» in Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on, 2010, pp. 224-231.

[3] S. Gebert, R. Pries, D. Schlosser, and K. Heck, Internet access traffic measurement and analysis: Springer, 2012.

[4] M. Mohamed, M. Amziani, D. Belaïd, S. Tata, and T. Melliti, «An autonomic approach to manage elasticity of business processes in the Cloud,» Future Generation Computer Systems, 2014.

[5] M. Ghobaei-Arani, S. Jabbehdari, and M. A. Pourmina, «An autonomic approach for resource provisioning of cloud services,» Cluster Computing, pp. 1-20, 2016.

[6] S. Islam, J. Keung, K. Lee, and A. Liu, «Empirical prediction models for adaptive resource provisioning in the cloud,» Future Generation Computer Systems, vol. 28, pp. 155-162, 2012.

[7] J. Huang, C. Li, and J. Yu, «Resource prediction based on double exponential smoothing in cloud computing,» in Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on, 2012, pp. 2056-2060.

[8] H. R. Qavami, S. Jamali, M. K. Akbari, and B. Javadi, «Dynamic Resource Provisioning in Cloud Computing: A Heuristic Markovian Approach,» in Cloud Computing, ed: Springer, 2014, pp. 102-111.

[9] M. Fallah, M. G. Arani, and M. Maeen, «NASLA: Novel Auto Scaling Approach based on Learning Automata for Web Application in Cloud Computing Environment,» International Journal of Computer Application, vol. 117, pp. 18-23, 2015.

[10] G. Moltó, M. Caballer, and C. de Alfonso, «Automatic memory-based vertical elasticity and oversubscription on cloud platforms,» Future Generation Computer Systems, vol. 56, pp. 1-10, 2016.

[11] A. G. García, I. B. Espert, and V. H. García, «SLA-driven dynamic cloud resource management,» Future Generation Computer Systems, vol. 31, pp. 1-11, 2014.

[12] S. Singh and I. Chana, «Q-aware: Quality of service based cloud resource provisioning,» Computers & Electrical Engineering, vol. 47, pp. 138-160, 2015.

[13] M. S. Aslanpour and S. E. Dashti, «SLA-aware resource allocation for application service providers in the cloud,» in 2016 Second International Conference on Web Research (ICWR), 2016, pp. 31-42.

[14] A. A. Bankole and S. A. Ajila, «Cloud client prediction models for cloud resource provisioning in a multitier web application environment,» in Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on, 2013, pp. 156-161.

[15] S. A. Ajila and A. A. Bankole, «Cloud client prediction models using machine learning techniques,» in Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual, 2013, pp. 134-142.

[16] P. D. Kaur and I. Chana, «A resource elasticity framework for QoS-aware execution of cloud applications,» Future Generation Computer Systems, vol. 37, pp. 14-25, 2014.

[17] R. Buyya, R. Ranjan, and R. N. Calheiros, «Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities,» in High Performance Computing & Simulation, 2009. HPCS›09. International Conference on, 2009, pp. 1-11.

[18] A.-F. Antonescu and T. Braun, «Simulation of SLA-based VM-scaling algorithms for cloud-distributed applications,» Future Generation Computer Systems, 2015.

[19] J. Li, S. Su, X. Cheng, M. Song, L. Ma, and J. Wang, «Cost-efficient coordinated scheduling for leasing cloud resources on hybrid workloads,» Parallel Computing, vol. 44, pp. 1-17, 2015.

[20] S. E. Dashti and A. M. Rahmani, «Dynamic VMs placement for energy efficiency by PSO in cloud computing,» Journal of Experimental & Theoretical Artificial Intelligence, pp. 1-16, 2015.

[21] N. R. Herbst, N. Huber, S. Kounev, and E. Amrehn, «Self-adaptive workload classification and forecasting for proactive resource provisioning,» Concurrency and computation: practice and experience, vol. 26, pp. 2053-2078, 2014.

[22] M. D. de Assunção, C. H. Cardonha, M. A. Netto, and R. L. Cunha, «Impact of user patience on auto-scaling resource capacity for cloud services,» Future Generation Computer Systems, vol. 55, pp. 41-50, 2016.