ADVISE – چارچوبی برای ارزیابی رفتار کشش سرویس ابری

چکیده

سرویس های ابری پیچیده برای مواجهه با حجم کارها و تغییرات مورد نیاز پویا بر فرآیندهای کنترل مختلف کشش متکی هستند. با این حال، اجرای فرآیند کنترل کشش در یک سرویس ابری به دلیل پیچیدگی ساختار سرویس، راهبردهای استقرار، و پویایی اصولی زیرساخت، از نظر کیفیت یا هزینه همیشه منجر به بهره وری بهینه نمی شود. بنابراین توانایی پیشین در برآورد و ارزیابی رابطه بین رفتار کشش سرویس ابری و فرآیندهای کنترل کشش برای انتخاب زمان اجرای فرآیند های مناسب کنترل کشش حیاتی است. در این مقاله، ADVISE را با چارچوبی برای ارزیابی و برآورد رفتار کشش سرویس ابری ارائه می کنیم. ADVISE ، ساختار سرویس، استقرار، زمان اجرای سرویس، فرآیندهای کنترل و اطلاعات زیرساخت ابری را جمع آوری می کند. بر اساس این اطلاعات، ADVISE از تکنیک های دسته بندی استفاده می کند تا رفتار کشش ابری که با کنترل کشش ایجاد شده را شناسایی کند. آزمایش های ما نشان می دهد که ADVISE می تواند رفتار کشش مورد انتظار بموقع را برای سرویس های مختلف ابری برآورد کند در نتیجه ابزار مفیدی برای کنترل کننده های کشش در بهبود کیفیت تصمیمات زمان اجرای کنترل کشش می شود.

1. مقدمه

یکی از ویژگی های کلیدی که باعث محبوبیت رایانش ابری می شود کشش است که در پاسخ به زمان اجرای نوسان حجم کارها، توانایی سرویس های ابری برای به دست آوردن و راه اندازی منابع بنا به تقاضا است. از دیدگاه مشتری، منابع با مقیاس بندی خودکار ابری می توانند زمان اجرای کار را به حداقل برسانند، بدون اینکه از بودجه اختصاص

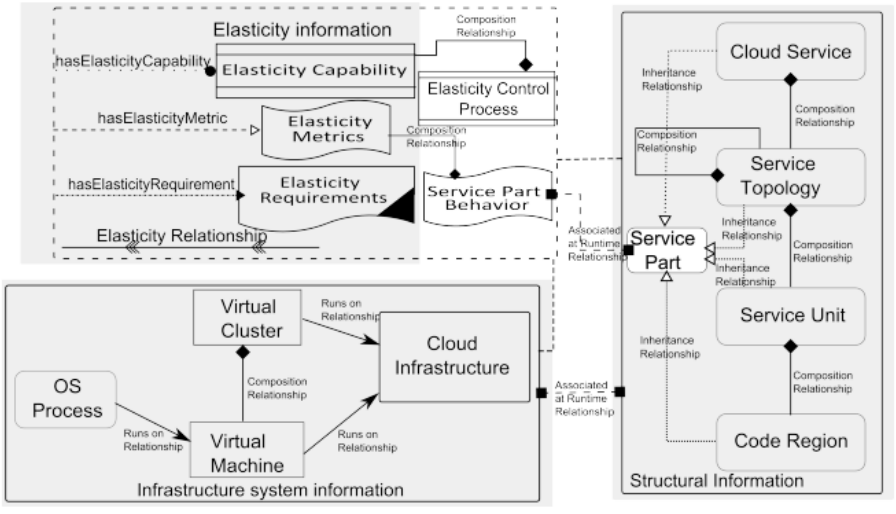
یافته بیشتر شود. از دیدگاه ارائه دهندگان ابری، تامین کشش کمک می کند تا بهره وری مالی آنها به حداکثر برسد هنگامیکه مشتریان خود را راضی نگه می دارند و هزینه های اجرایی را کاهش می دهند. با این حال، تأمین خودکار کشش یک کار ساده و جزیی نیست.

هنگامی که یک آستانه متریک دچار اختلال می شود رویکرد معمول که توسط بسیاری از کنترل کننده های کشش استفاده می شود (1، 2) برای نظارت سرویس ابری و تامین نمونه های مجازی است. این رویکرد ممکن است برای مدل های ساده سرویس کافی باشد، اما با در نظرگیری سرویس های ابری توزیع شده با مقیاس بزرگ با وابستگی های متقابل مختلف، درک عمیق تر از رفتار کشش ضروری می شود. به همین دلیل، کار موجود [2، 3] تعدادی از فرآیندهای کنترل کشش را برای بهبود عملکرد و کیفیت سرویس ابری شناسایی کرده است، حال آنکه به طور اضافی تلاش در کاهش هزینه می کند. با این حال هنوز یک سوال مهم بدون پاسخ باقی مانده است: مناسب ترین فرآیند کنترل کشش برای یک سرویس ابری در زمان اجرای یک وضعیت خاص کدام است؟ هم مشتریان و هم ارائه دهندگان ابری می توانند از اطلاعات روشنی بهره ببرند مانند اینکه چگونه اضافه کردن نمونه جدید به یک سرویس ابری بر حاصل کار استقرار کل و بر هر بخشی از سرویس ابری بصورت جداگانه تاثیر خواهد گذاشت. بدین ترتیب، آگاهی از رفتار کشش سرویس ابری تحت کنترل و حجم کارهای مختلف، برای کنترل کننده های کشش در بهبود تصمیم گیری زمان اجرا بسیار مهم است.

برای این منظورطیف گسترده ای از رویکردها متکی هستند بر نمایه سازی سرویس و یا یادگیری از اطلاعات گذشته [3-5] که پیشنهاد شده است. با این حال، این رویکردها تصمیمات خود را فقط برای ارزیابی متریک های سطح پایین VM(به عنوان مثال استفاده از CPU و حافظه) محدود می کنند و از تصمیمات کشش مبتنی بر رفتار سرویس ابری در چندین سطح پشتیبانی نمی کنند (به عنوان مثال، در هر گره، لایه، کل سرویس). علاوه بر این، رویکردهای کنونی فقط بکارگیری از منابع را بدون در نظر گرفتن کشش به عنوان یک ویژگی چند بعدی متشکل از سه بعد (هزینه، کیفیت، و کشش منابع) ارزیابی می کنند. در نهایت رویکردهای موجود، نتیجه فرایند کنترل بر سرویس کلی را در نظر نمی گیرند جاییکه اجرای یک فرآیند کنترل در بخش اشتباه سرویس ابری می تواند منجر به

اثرات جانبی مانند افزایش هزینه یا کاهش عملکرد سرویس کل شود. در کار قبلی که انجام دادیم، با مفاهیم فضا و مسیر کشش (6) بر رفتار پیشین و کنونی مدلسازی یا استفاده از الگوریتم های مختلف برای تعیین زمان اجرا در رفتار مشاهده شده تمرکز کردیم اما بدون اینکه رفتار مورد انتظار از بخش های مختلف سرویس را مدلسازی کنیم (به عنوان مثال با تعیین نقطه تغییر).

در این مقاله بر عنوان کردن محدودیت های فراتر از چارچوب ADVISE (ارزیابی رفتار کشش سرویس ابری) تمرکز می کنیم که رفتار کشش سرویس ابری را با استفاده از انواع مختلف اطلاعات مانند ساختار سرویس، راهبرد های استقرار و زیرساخت های اصولی پویا در هنگام استفاده از محرک های مختلف خارجی برآورد می کند (به عنوان مثال، فرایندهای کنترل کشش). در هسته ADVISE یک فرآیند ارزیابی مبتنی بر دسته بندی است که از این نوع اطلاعات برای محاسبه رفتار کشش مورد انتظار در بخش های مختلف سرویس استفاده می کند. برای ارزیابی اثربخشیADVISE ، آزمایش ها بر یک بستر ابری عمومی با یک تست بد testbed متشکل از دو سرویس ابری متفاوت انجام گرفت. نتایج نشان می دهد که ADVISE رفتار کشش مورد انتظار را برای سرویس های مختلف با برآورد میزان خطای کم حاصل می‌کند. ADVISE همراه با کنترل کننده های کشش خود می تواند با ارائه دهندگان ابری یکی شود تا کیفیت تصمیم خود را بهبود بخشد و یا توسط ارائه دهندگان سرویس ابری استفاده شود تا ارزیابی و درک کند چگونه فرآیندهای مختلف کنترل کشش بر سرویس خود تاثیر قرار می گذارند.



شکل 1 قابلیت های کشش که برای اهداف کشش مختلف در نظر گرفته شده

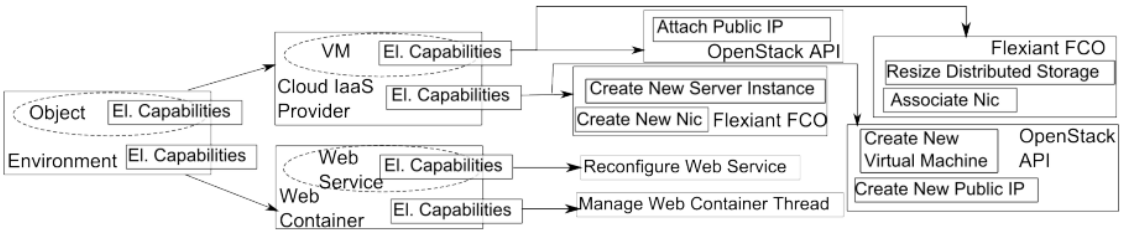
بقیه این مقاله به شرح زیر است: در بخش 2 اطلاعات مربوط به سرویس ابری ارائه می شود. در بخش 3 فرایند ارزیابی رفتار کشش را ارائه می کنیم. در بخش 4 اثربخشی چارچوب ADVISE را ارزیابی می کنیم. در بخش 5 درباره کار مرتبط صحبت می کنیم. بخش 6 این مقاله را به پایان می رساند.

2. ساختار سرویس ابری و اطلاعات زمان اجرا

2.1 اطلاعات سرویس ابری

در این مطالعه دنباله شرح سرویس موجود [7] در برنامه ابری را بگونه سرویس ابری اشاره می کنیم. یک سرویس ابری را می توان به توپولوژی های سرویس تقسیم کرد (مثلا رشته ایی از تجارت یا بخشی از یک جریان کاری) که گروهی از واحدهای خدماتی بصورت لغوی مربوط به هم را نشان می دهد. یک واحد خدماتی (به عنوان مثال سرویس وب) یک ماژول با ارائه قابلیت محاسبات یا داده را نشان می دهد. به منظور اینکه به این ساختارهای سرویس ابری در سطح جهانی اشاره کنیم از اصطلاح بخش های سرویس (SP) استفاده می کنیم.

مدل سرویس ابری مفهومی که در [8] ارائه شده را با مجموعه غنی از انواع اطلاعات برای تعیین رفتار کشش ابری گسترش می دهیم. شکل 1، گسترش های ما را (پس زمینه سفید) در شامل کردن فرآیندهای کنترل کشش، رفتارهای بخش سرویس، و بخش های سرویس نشان می دهد. به طور کلی، این نمایه شامل این موارد است (i) اطلاعات ساختاری در توصیف ساختار معماری برنامه ای که در ابر اجرا می شود، (ii) اطلاعات سیستم زیرساختی که اطلاعات زمان اجرا در مورد منابع اختصاص یافته از بستر اصولی به سیستم ابری را توصیف می کند، و (iii) اطلاعات کشش، که برای توصیف متریک ها، الزامات، و قابلیت های کشش، هم با اطلاعات سیستم ساختاری و هم غیرساختاری مرتبط است.



شکل 2 قابلیت های کشش که برای اهداف کشش مختلف در نظر گرفته شده

اطلاعات کشش، از متریک های کشش، الزامات کشش، و قابلیت های کشش تشکیل شده است که هر کدام از آن ها به منابع مختلف زیرساخت ها یا بخش های سرویس مرتبط هستند. قابلیت های کشش با هم به عنوان فرایندهای کنترل کشش (ECPs) دسته بندی می شوند همانطور که در بخش بعدی شرح داده می شوند، و رفتارهای خاص کشش را بر بخش های مختلف سرویس تحمیل می کنند که آن را به عنوان رفتارهای بخش سرویس مدل می کنیم. رفتارهای بخش سرویس را مدل می کنیم، زیرا کنترل کننده ها باید تأثیر اجرای ECP در سطوح مختلف را تعیین کنند (به عنوان مثال قبل از اینکه گره پایگاه داده جدیدی تخصیص دهیم، تاثیر در توپولوژی سرویس پایگاه داده و در کل سطح سرویس ابری باید تعیین شود). به طور مفهومی، رفتار بخش سرویس به عنوان رفتار SP_i برای یک SP_iخاص در یک دوره تعریف شده از زمان [شروع، پایان] با تمام متریک های $M_a^{S\ Pi}$ نشان داده می شود که SP_i را نظارت می کند. بنابراین، رفتار یک سرویس ابری که به عنوان $Behavior_{CloudService}$ مشخص شده است در طول یک دوره زمانی به عنوان مجموعه ای از همه رفتارهای SP سرویس ابری تعریف می شود:

$$M_a^{SP_i}[start, end] = \{M_a(t_j)|SP_i \in ServiceParts, j = \overline{start, end}\} \quad (1)$$

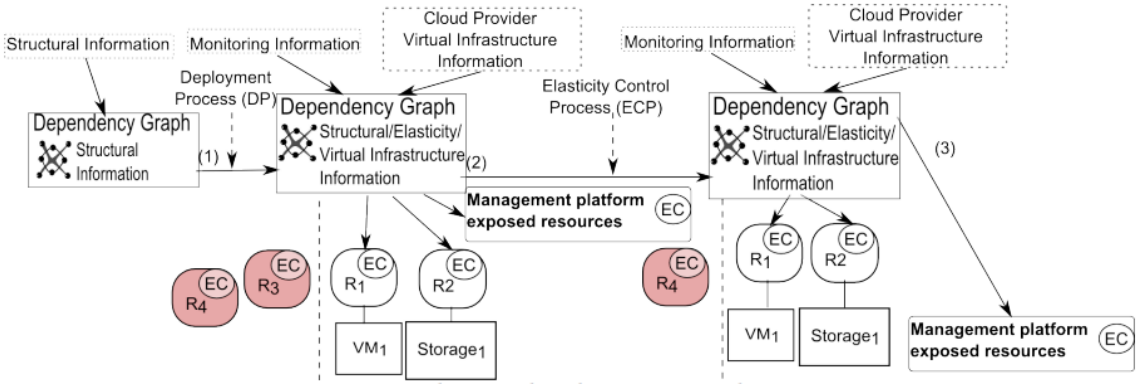
$$Behavior_{SP_i}[start, end] = \{M_a^{SP_i}[start, end]|M^a \in Metrics(SP_i)\} \quad (2)$$

$$Behavior_{CloudService}[start, end] = \{Behavior_{SP_i}[start, end]|SP_i \in ServiceParts(CloudService)\} \quad (3)$$

اطلاعات فوق از طریق یک نمودار وابستگی کشش در زمان اجرا به دست می آید و آن نیز مدیریت می شود که دارای مفاهیم گره از مدل ارائه شده در شکل 1 (به عنوان مثال، ماشین مجازی) و روابط (مثلا رابطه ی کشش) است. نمودار وابستگی کشش محصور شده است و به طور مستمر با (i) اطلاعات پیش استقرار، مانند توصیف توپولوژی سرویس (مثلا TOSCA {7}) یا نمایه سازی اطلاعات; و (ii) اطلاعات زمان اجرا مانند مقادیر متریک از ابزارهای نظارت و یا اطلاعات منابع اختصاص یافته از ارائه دهندگان ابری API به روز می شود.

2.2 فرایندهای کنترل کشش

قابلیت های کشش (EC) مجموعه عملیات مربوط به یک سرویس ابری هستند که یک متولی سرویس ابری (به عنوان مثال کنترل کننده کشش) ممکن است از آن استفاده کند و بر رفتار سرویس ابری تاثیر بگذارد. چنین قابلیت هایی می تواند برای این موارد قرار داده شود: (i) بخش های مختلف سرویس (ii) ارائه دهندگان ابر، یا (iii) منابعی که توسط ارائه دهندگان ابر تامین می شود. یک EC را می توان به عنوان نمایش انتزاعی API در نظر گرفت، که در میان ارائه دهندگان و سرویس های ابری متفاوت است. شکل 2، زیرمجموعه های مختلفی از EC ها که برای کاربرد وب نمونه ایی ایجاد شده را نمایش می دهد هنگامیکه بر دو بستر مختلف ابری قرار می گیرد (به عنوان مثال، ابر خصوصی Flexiant و Openstack)، و همچنین EC هایی برای سرویس ابری و نرم افزار نصب شده قرار داده می شوند. سرویس ابری در هر دو بستر ابری ذکر شده باید در محیط خاص اجرا شود (به عنوان مثال، وب سرور Apache Tomcat)، و تمام این قابلیت ها، زمانی که توسط یک کنترل کننده کشش اجرا می شود، بر روی بخش های مختلف سرویس ابری تاثیر خواهد گذاشت. به عنوان مثال، قابلیت های کشش یک توپولوژی وب سرور که از سرویس ابری است حتی اگر در اولین نگاه مشهود نباشد می تواند بر عملکرد عقبه backend پایگاه داده آن تاثیر بگذارد.



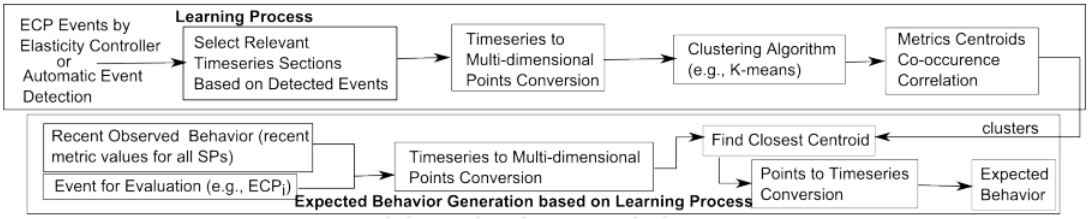
شکل 3 سیر تکامل سرویس ابری کشش

2.3 کشش سرویس ابری در زمان اجرا

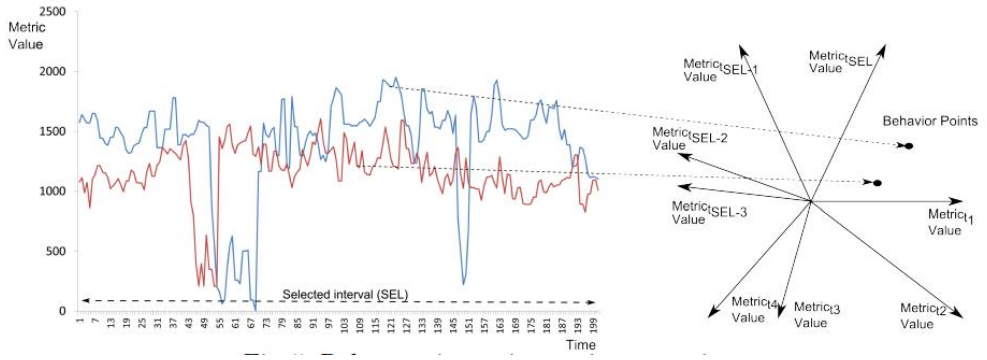
برای اینکه قادر باشیم اثرات ECP ها بر بخش های سرویس را برآورد کنیم بر نمودار وابستگی کشش تکیه می کنیم که تمام متغیرهایی که به سیر تکامل رفتار کشش سرویس ابری کمک می کنند را ثبت می کند. سمت چپ شکل 3، سرویس ابری در زمان پیش استقرار را نشان می دهد جایی که کنترل کننده های کشش خودکار، فقط از اطلاعات ساختاری که توسط منابع مختلف ارائه شده است در مورد آن آگاهی دارند (از جمله توضیحات سرویس TOSCA). پس از اجرای فرایند استقرار (به عنوان مثال، ایجاد ماشین x و پیکربندی نرم افزارz) نمودار وابستگی کشش بطور اضافی حاوی اطلاعات مربوط به زیرساخت های به دست آمده از ارائه دهنده ابری، و اطلاعات کشش بدست آمده از نظارت کردن سرویس ها با نشان دادن سیر تکامل متریک برای SP های مختلف خواهد شد. این اطلاعات به طور مستمر در طول زمان اجرا به روز می شود (مرحله 3 در شکل 3)، از این رو برای برآورد رفتار فرض می کنیم که اطلاعات کاملی داریم (یعنی هیچ اطلاعاتی کم نداریم).

منابع زیرساختی همانطور که قبلا ذکر شد با قابلیت های کشش (ECدر شکل 3) مرتبط است که تغییراتی را که باید اعمال شود و مکانیسم های راه اندازی آنها را (مانند APIمختص EC) توصیف می کند. علاوه بر این یک بستر ابری، ECها را برای ایجاد منابع جدید یا ارائه سرویس های جدید قرار می دهد (به عنوان مثال، افزایش حافظه یک ECاست که برای VM قرار گرفته از این رو ایجاد VMجدید، یک EC است که برای بستر ابری قرار گرفته است).

در این زمینه، برای کشف اثراتی که ECP بوجود می آورد، همبستگی بین متریک ها در هر بخش سرویس در نظر گرفته می شود که از نمودار وابستگی کشش برای آن استفاده می کنیم. این اطلاعات را تجزیه و تحلیل می کنیم تا تأثیر یک ECP در همه بخش های سرویس را مشخص کنیم، صرف نظر از این که آیا ECP یک برنامه خاص است یا هیچ پیوند آشکاری به دیگر بخش های سرویس ندارد. در حقیقت، همانطور که در بخش 4 نشان داده شده است، تأثیر ECP های مختلف بر بخش های مختلف و در کل سرویس ابری بسیار قابل توجه است.



شکل 4 مدلسازی فرآیند رفتار کشش ابری



شکل 5 بخش های سری زمانی مربوطه در نقاط

3. ارزیابی رفتار کشش سرویس ابری

راه حل های موجود که برای یادگیری مدل های رفتار [4،5] است، مدل های متریک گسسته را بدون همبستگی آنها با متغیرهای متعدد که بر رفتار سرویس ابری تاثیر می گذارد یاد می دهد. در مقابل آنها، در حال یادگیری رفتار بخش های سرویس ابری مختلف و ارتباط آنها با ECP های مختلف هستیم البته نه تنها با آنهایی که به طور مستقیم مرتبط هستند، همچنین تاثیر یک ECP را با توجه به همبستگی های بین چند متریک و در میان چندین بخش سرویس برآورد می کنیم. فرآیند یادگیری که برای تعیین رفتار بخش سرویس ابری استفاده شده است در شکل 4 نشان داده شده، و با تصحیح پایگاه آگاهی که قبلا جمع آوری شده بطور پیوسته اجرا می شود.

3.1 فرایند یادگیری

پردازش داده های ورودی. فرایند یادگیری ما به گونه ورودی هر سیر تکامل متریک $M_a^{S\ Pi}$(شروع، اکنون) (معادله 3 را ببینید) از آغاز اجرای سرویس بر بستر ابری کنونی است. برای اینکه سیر تکامل مورد انتظار متریک ها

در پاسخ به اجرای یک ECP خاص را برآورد کنیم برای هر متریک نظارت شده از هر بخش سرویس، یک سری زمانی مربوطه (RT S) انتخاب می کنیم تا آن را با $M_a^{S\ Pi}$ قبلی مقایسه کنیم (شروع، اکنون). اندازه RTS به شدت به میانگین زمانی لازم برای اجرای ECP بستگی دارد (نگاه کنید به بخش 4.3). در نتیجه، متریک RTS یک زیر دنباله $M_a^{S\ Pi}$ از قبل از اجرای ECP است تا زمانی که اجرا کامل می شود:

$$RTS_{M_a}^{SP_i} = M_a^{SP_i}[x - \frac{\delta + ECP_{time}}{2}, x + \frac{\delta + ECP_{time}}{2}], \quad (4)$$
$$[ECP_{startTime}, ECP_{endTime}] \subset [x - \frac{\delta + ECP_{time}}{2}, x + \frac{\delta + ECP_{time}}{2}]$$

، جایی که x نشان دهنده ECP و S طول دوره ای است که برای ارزیابی آن تلاش می کنیم.

$$BP_a^{SP_i}[j] = RTS_{M_a}^{SP_i}[t(j)], j = 0, ..., n, BP : M^{SP} \mapsto R^n, n = \delta + ECP_{time} \ (5)$$

بگونه مرحله پیش پردازش ورودی ، S + ECPtime را به عنوان نقاط چند بعدی، BPدر معادله 5 و در فضای اقلیدسی n بعدی نشان می دهیم (نگاه کنید به شکل 5)، جایی که مقدار ابعاد t (j) برچسب زمان j از RTS کنونی است.

فرایند دسته بندی. برای اینکه رفتار مورد انتظاردر نتیجه ی اجرایECP را شناسایی کنیم دسته های Clusterspi نقاط رفتاری را برای همه بخش های سرویس و برای هر ECP مبتنی بر فاصله بین نقاط رفتار طرح ریزی می کنیم همانگونه که در معادله 6 بیان شده است. رویکرد خود را تنها با در نظر گرفتن ECPموجود برای SPiکنونی محدود نمی کنیم، همانطور که قبلا ذکر شد اجرای ECP در بخش سرویس خاص ممکن است بر رفتار یک SP دیگر یا سرویس ابری کل تاثیر بگذارد. تابع هدف این فرآیند یافتن نقطه رفتار چند بعدی C (Θ *) است، که فاصله بین نقاط متعلق به چنین دسته ایی را به حداقل می رساند (نگاه کنید به معادله 7). از آنجا که تمرکز این مقاله ارزیابی کیفیت الگوریتم های دسته بندی مختلف نیست، از الگوریتم K-means استفاده کنیم، به این ترتیب جاییکه تعداد دسته ها $K = \sqrt{N/2}$ است Nتعداد اهداف می شود. با این حال همانطور که در بخش 4 نشان داده شده است، حتی با الگوریتم ساده K-Means، رویکرد ما رفتار کشش مورد انتظار را با برآورد میزان خطا پایین حاصل می کند.

$$dist(BP_a^x, BP_a^y) = \sqrt{\sum_i (BP_a^x[i] - BP_a^y[i])^2} \quad (6)$$

$$\Theta^* = \arg\min \sum_{k=0}^{K} \sum_{i=0}^{N} \theta_{i,k} dist(Cluster_k, BP_i), \quad \theta_{i,k} = \begin{cases} 1 & BP_i \in Cluster_k \\ 0 & BP_i \notin Cluster_k \end{cases} \quad (7)$$

پس از اینکه دسته های نقطه ای با ابعاد S + _ECPtime را بدست آوردیم برای هر SPi یک ماتریس همبستگی GMspi [C_x, C_y]، که Cx مرکز دسته x است را ایجاد می کنیم و احتمال می دهیم همه متریک های مختلف از دسته ها با هم ظاهر شوند (به عنوان مثال، افزایش قابلیت اطمینان داده ها معمولا با افزایش هزینه مرتبط است). یک بخش در CM نمایانگر نسبت بین تعداد نقاط رفتار Cx و Cy است که با هم در جهت تعداد کل نقاط رفتار درنظر گرفته می شوند. زمانی که نقاط رفتار از یک دسته به دسته دیگر حرکت می کند، یا زمانی که ECP های جدید اجرا می شوند این ماتریس بطور مداوم به روز می شود، از این رو پایگاه آگاهی افزایش می یابد.

3.2 تعیین رفتار کشش مورد انتظار

در مرحله ی ایجاد رفتار موردانتظار مبتنی بر فرایند یادگیری که در شکل 4 نشان داده شده است آخرین مقدار متریک ها را برای هرSPi یعنی $M^{s}{}_{a}{}^{Pi}$ (کنونی - s، کنونی] انتخاب می کنیم و £ ECP که کنترل کننده است برای اجرا و یا برای آن که کاربر می خواهد از این اثرات آگاهی پیدا کند در نظر گرفته می شود. رفتار مورد انتظار را دریافتیم (معادله 8 را ببینید) که شامل یک مرکز دسته چندتایی از دسته های ساخته شده در طول فرآیند یادگیری است که در بخشی از سرویس ابری به رفتار متریک کنونی خیلی نزدیک است که ما در حال تمرکز بر آنها هستیم زیرا با هم در سراسر اجرای سرویس ابری ظاهر می شوند. نتیجه این مرحله، برای هر متریک SPi، لیستی از مقادیر مورد انتظار از اجرای £ ECP است (به عنوان مثال مقادیر مورد انتظار از هر متریک برای این مورد که کاربر می خواهد یک سرویس جدید وب از نوع x را در همان محفظه برنامه وب قرار دهد).

$$\{C_{i_{a^1}}^{M_{a^1}}, ..., C_{i_{a^m}}^{M_{a^m}} | M_{a^m} \in Metrics(SP_i)\} \quad (8)$$

4. آزمایش ها

برای اینکه اثربخشی رویکرد پیشنهادی را ارزیابی کنیم چارچوب ADVISE که شامل مفاهیمی است که قبلا شرح داده شده را توسعه داده ایم. نسخه ADVISE کنونی انواع مختلفی از اطلاعات را برای پر کردن نمودار وابستگی کشش جمع آوری می کند مانند (i) :اطلاعات ساختاری، از توصیف سرویس TOSCA ؛ (ii) اطلاعات عملکرد کاربردی و زیرساخت از سیستم های نظارت JCatascopia (9) وMELA (6) (iii) اطلاعات کشش مربوط به ECPاز کنترل کننده کشش rSYBL (8) که در آن یک پلاگین اجرایی را گسترش دادیم تا ECP ها به طور تصادفی بر روی سرویس های ابری اجرا شوند. برای اینکه عملکرد چارچوبADVISE را ارزیابی کنیم تست بدی testbed متشکل از دو سرویس مستقر در ابر عمومی Flexiant ایجاد کردیم. در هر دو سرویس ابری، ECPهای تصادفی را که در بخش های مختلف سرویس قرار گرفته اند اجرا می کنیم. از کنترل کننده منطقی استفاده نمی کنیم، زیرا علاقه مند به برآورد رفتار کشش تمام بخش های سرویس در نتیجه ی اجرای تصمیم کنترل کشش خوب و بد هستیم.

سرویس ابریECP		**توالی**
عملیات		
مقیاس پذیری عمودی لایه سرور کاربردی: (i) قطع سرویس پخش ویدیو (ii) حذف نمونه از HAproxy (iii) راه اندازی مجدد HAproxy (iv) قطع علمل نظارت JCatascopia (v) حذف نمونه ها	ECP_1	سرویس ویدیویی
مقیاس پذیری افقی لایه سرور کاربردی(i) ایجاد رابط شبکه جدید (ii) نمونه سازی ماشین مجازی جدید (iii) استقرار و پیکربندی سرویس پخش ویدیو (iv) استقرار و آغاز عامل نظارت JCatascopia (v) اضافه کردن نمونه ip به HAproxy (vi) راه اندازی مجدد HAproxy	ECP_2	
مقیاس پذیری عمودی عقبه ذخیره سازی ویدیو توزیع شده (i) انتخاب نمونه برای حذف کردن (ii) انهدام داده های نمونه در سایر گره ها (با استفاده از ابزار گره Cassandra (iii) قطع عامل نظارت JCatascopia (iv) حذف نمونه	ECP_3	
مقیاس پذیری افقی عقبه ذخیره سازی ویدیو توزیع شده (i) ایجاد رابط شبکه جدید (ii) نمونه سازی نمونه جدید (iii) استقرار و پیکربندی Cassandra (برای مثال تخصیص نشانه برای گره) (iv) استقرار و راه اندازی عامل نظارت JCatascopia (v) راه اندازی Cassandra	ECP_4	
مقیاس پذیری عمودی واحد خدماتی پردازش رخداد (i) حذف سرویس از HAproxy (ii) راه اندازی مجدد HAproxy (iii) حذف ماشین مجازی بطور بازگشتی	ECP_5	M2M
مقیاس پذیری افقی واحد خدماتی پردازش رخداد (i) ایجاد رابط شبکه جدید (ii) ایجاد ماشین مجازی جدید (iii) اضافه کردن سرویس IP به فایل پیکربندی HAproxy	ECP_6	DaaS
مقیاس پذیری افقی واحد خدماتی گره داده (i) انهدام گره (کپی داده از ماشن مجازی حذف شده) (ii) حذف ماشین	ECP_7	

مجازی بطور بازگشتی		
مقیاس پذیری عمودی واحد خدماتی گره داده (i) ایجاد رابط شبکه جدید (ii) ایجاد ماشین مجازی (iii) تنظیم پورت ها (iv) تخصیص نشانه به گره (v) تنظیم کنترل کننده دسته (vi) راه اندازی Cassandra	ECP_8	

ADVISE بطور معمول اطلاعات نظارت را در دو فرمت دریافت می کند: (i) به عنوان فایل های ساده*.csv یا (ii) به طور خودکار اطلاعات نظارت را از MELA بدست می آورد. ADVISE می تواند هم در مرحله نمایه سازی / پیش استقرار یا در زمان اجرا برای انواع مختلف سرویس مورد استفاده قرار گیرد یعنی هر زمان که اطلاعات نظارت و ECP های اجرا شده، برای ایجاد برآورد متریک های مختلف از بخش های سرویس قابل دسترس هستند.

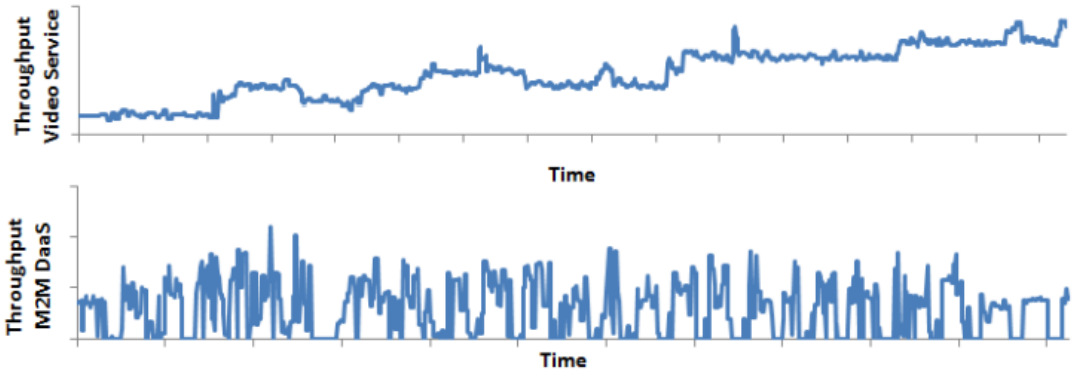
4.1 سرویس های آزمایشی

اولین سرویس ابری، یک برنامه وب سه لایه است که به کاربران آنلاین سرویس پخش ویدئو را ارائه می دهد: (i) یک متعادل کننده بار HAProxy که درخواست مشتری (یعنی دانلود یا آپلود ویدئو) را در سرتاسر سرورهای کاربردی توزیع می کند: (ii) یک لایه سرور کاربردی، که در آن هر سرور کاربردی یک سرور Apache Tomcat است که حاوی سرویس وب پخش ویدئو می باشد؛ (iii)یک عقبه ذخیره سازی اطلاعات توزیع شده Cassandra NoSQL که محتوای ویدئوی مورد نیاز را بازیابی می کند. چارچوب ADVISE را با ایجاد درخواست مشتری تحت نرخ پایدار ارزیابی کرده ایم، جایی که بار به نوع درخواست ها و اندازه ویدئوی درخواست شده بستگی دارد که آن در الگوی حجم کاری در شکل 6 نشان داده شده است.

سرویس دوم در ارزیابی ما یک DaaS ماشین به ماشین (M2M) است که اطلاعات نشات گرفته از چندین نوع مختلف از حسگرهای داده را (از جمله دما، فشار جو یا آلودگی) پردازش می کند. به طور خاص، DaaS ماشین به ماشین متشکل از یک توپولوژی سرویس پردازش رخداد و یک توپولوژی سرویس پایان داده است. هر توپولوژی سرویس متشکل از دو واحد خدماتی است، یکی با هدف پردازش، و دیگری به عنوان متعادل کننده / کنترل کننده عمل می کند. برای تأکید بر این سرویس ابری، اطلاعات مربوط به رخداد حسگر را تصادفی ایجاد می کنیم (نگاه کنید به شکل 6) که آن را توسط توپولوژی سرویس پردازش رخداد بوجود می آوریم و از توپولوژی سرویس پایان

داده ذخیره می شود/بازیابی می شود. جداول 1 و 2 ECP های مربوط به هر بخش سرویس و متریک های تجزیه و تحلیل شده مربوط به دو سرویس ابری را فهرست می کند.

سرویس ابری	نام بخش سرویس متریک ها	
سرویس ویدیویی	لایه سرور کاربردی	هزینه، تعداد مشغول، بکارگیری حافظه، خروجی درخواست
	عقبه ذخیره سازی ویدیو توزیع شده	هزینه، کاربرد CPU، کاربرد حافظه، تاخیر جستجو
M2M DaaS	سرویس ابری	هزینه هر مشتری در هر ساعت (هزینه/مشتری/ h)
	توپولوژی سرویس پردازش رخداد	هزینه، زمان پاسخ، خروجی، تعداد مشتریان
	توپولوژی سرویس پایان داده	هزینه، تاخیر، کاربرد CPU



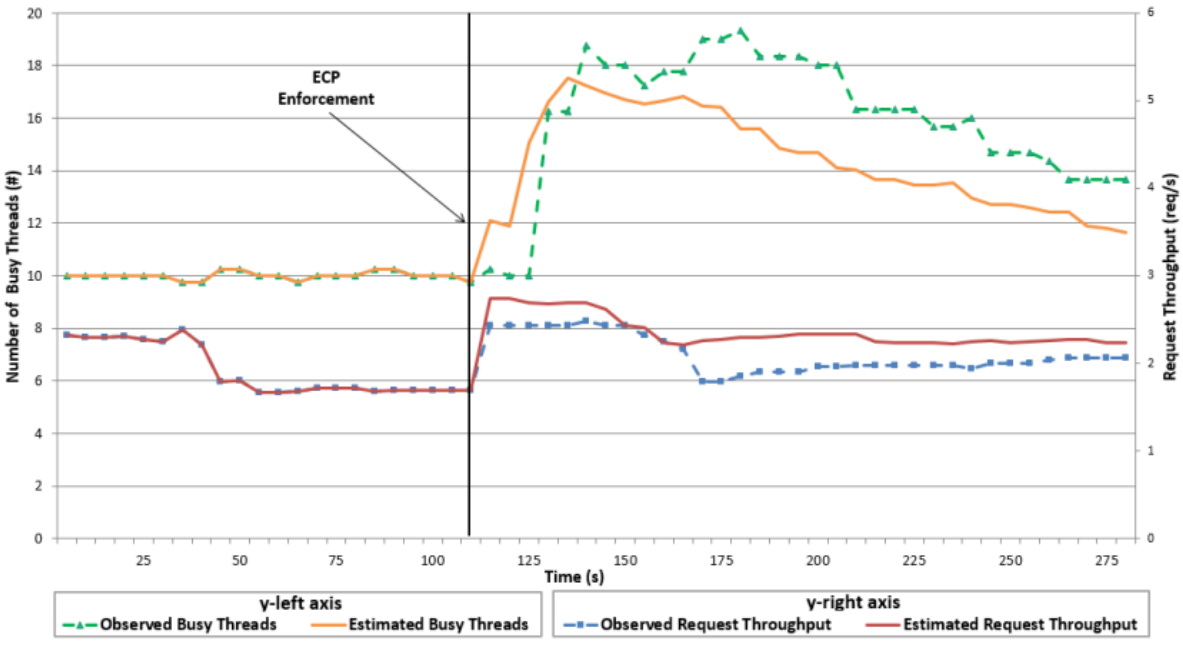
شکل 6 حجم کاری اعمال شده بر دو سرویس

4.2 برآورد رفتار کشش

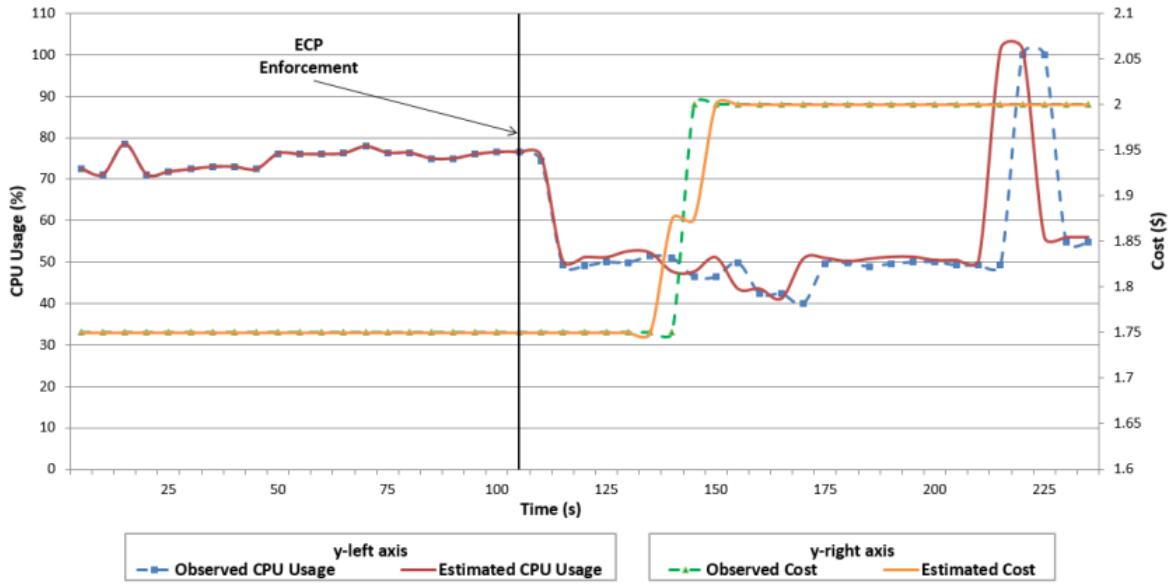
سرویس آنلاین پخش ویدئو. شکل 7 زمانی که یک سرور کاربردی از لایه ECP رخ می دهد هم رفتار مشاهده شده و هم برآورد شده را برای لایه سرور کاربردی از سرویس ابری نشان می دهد (ECP). در ابتدا، مشاهده می کنیم که میانگین خروجی درخواست در هر سرور کاربردی کاهش می یابد. این به دلیل دو مورد احتمالی است: (i) عقبه ذخیره سازی ویدئو تحت آماده شدن است و نمی تواند تعداد فعلی درخواست ها را برآورده کند، که به نوبه خود باعث می شود که درخواست ها در صف قرار بگیرند؛ (ii) یک افت ناگهانی در درخواست های مشتری وجود دارد که حاکی از این است که سرورهای کاربردی به طور موثر مورد استفاده قرار نمی گیرند. متوجه می شویم که پس از

اینکه عمل مقیاس پذیری عمودی رخ می دهد، میانگین ورودی درخواست و تعداد خطوط مشغول افزایش می یابد که نشان دهنده این است که این رفتار مطابق با مورد دوم است در جاییکه منابع به طور موثر استفاده نمی شوند. ADVISEیک همبستگی روشنی بین دو متریک نشان داد تا متوجه شویم چه موقع تصمیم گیری شود که ECP برای این رفتار اجرا شود.

به طور مشابه، در شکل 8 هنگامی که عمل مقیاس پذیری افقی (با اضافه کردن گره Cassandra به حلقه) به علت استفاده زیاد از CPU رخ می دهد هر دو رفتار مشاهده شده و برآورد شده درعقبه ذخیره سازی ویدئوی توزیع شده را نشان می دهیم. همانطور که با برآورد نشان داده شده است مشاهده می کنیم پس از اینکه عمل مقیاس پذیری افقی رخ می دهد بکارگیری واقعی از CPU به مقدار عادی کاهش می یابد. در نهایت، از شکل 7 و 8 نتیجه می گیریم که برآورد ADVISE از الگوی رفتار واقعی با موفقیت پیروی می کند و در هر دو مورد، با گذشت زمان منحنی ها همگرا می شوند.

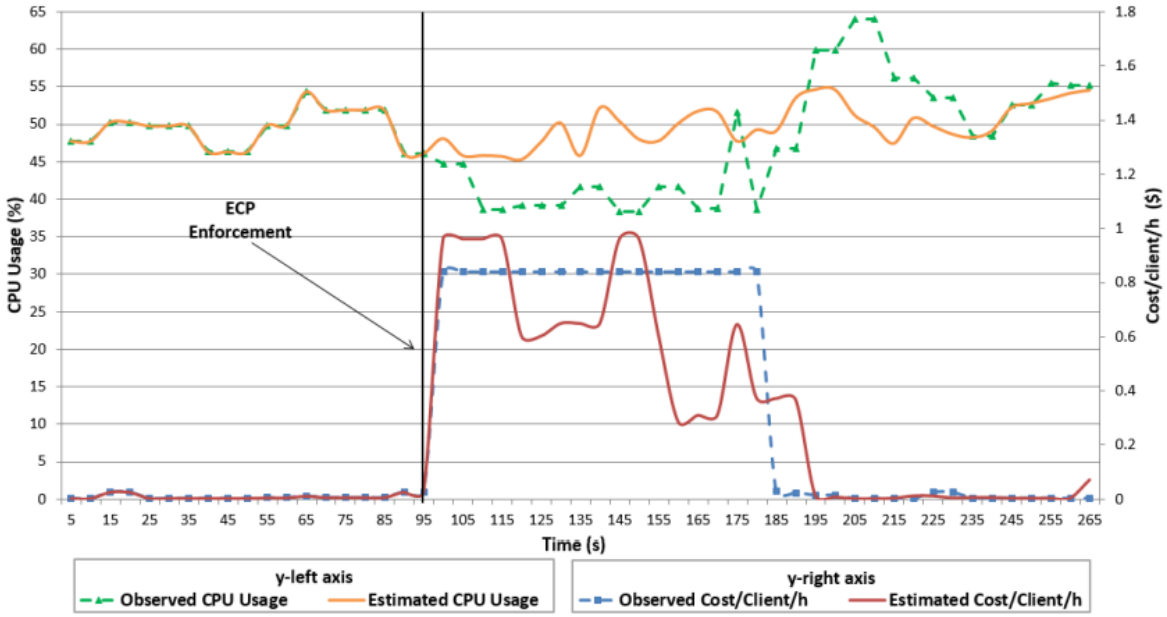


شکل 7 تاثیر ECP_1 بر لایه سرور کاربردی

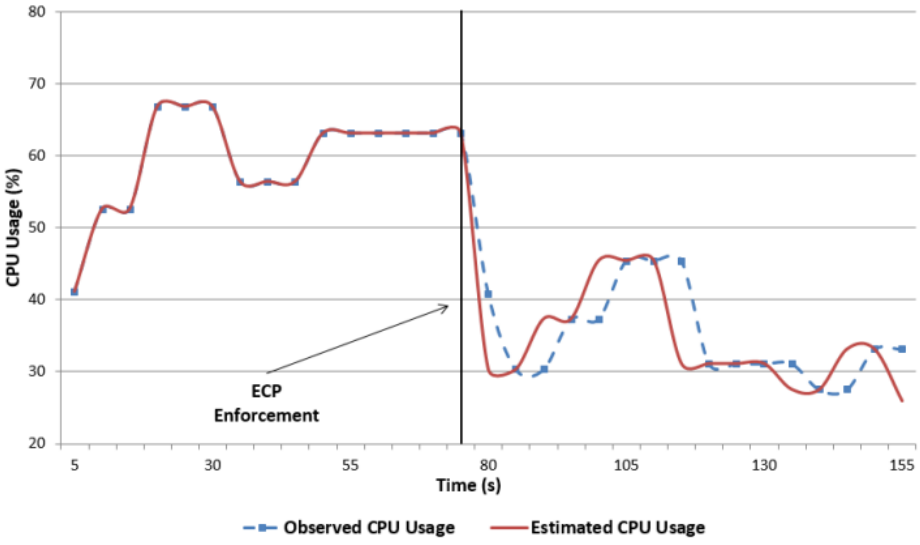


شکل 8 تاثیر ECP_4 بر کل سرویس پخش ویدیو

DaaS ماشین به ماشین در شکل 9 نشان می دهد که چگونه هدف گذاری ECP یک واحد خدماتی، بر کل سرویس ابری تاثیر می گذارد. هزینه /مشتری / h یک متریک پیچیده است (به جدول 2 مراجعه کنید) که نشان می دهد چقدر استقرار سرویس در سنجش تعداد فعلی کاربران سودمند است. گرچه هزینه / مشتری / h به دلیل نوسانات بالا در تعداد مشتریان به وضوح برآورد نشده است رویکرد ما تقریب می زند که چگونه سرویس ابری بر حسب زمان مورد انتظار و نوسانات متریک مورد انتظار رفتار می کند. این اطلاعات برای کنترل کنندگان کشش در بهبود تصمیمات خود مهم است هنگامیکه این ECP را اجرا می کنیم البته با دانستن اینکه چقدر هزینه / مشتری / h در کل سرویس ابری تحت تأثیر قرار خواهد گرفت. اگر چه استفاده از CPU به طور کامل برآورد نشده است چون آن یک متریک بسیار نوسانی است و به استفاده از CPU در هر سطح واحد خدماتی بستگی دارد، همچنین دانستن مبنایی از این متریک می تواند به تصمیم اینکه آیا این ECP مناسب است یا خیر کمک کند (به عنوان مثال برای برخی کاربردهای استفاده از CPU بیش از 90٪ برای یک دوره زمانی ممکن است غیرقابل قبول باشد).



شکل 9 تاثیر ECP_7 بر M2M DaaS



شکل 10 تاثیر ECP_8 بر واحد خدماتی کنترل کننده داده ها

ADVISE می تواند تاثیر ECP یک بخش سرویس را بر یک بخش سرویس متفاوت برآورد کند حتی اگر ظاهرا غیر مرتبط باشند. شکل 10 نشان می دهد برآوردی را در مورد اینکه چگونه واحد خدماتی کنترل کننده داده توسط داده های منتقل شده برای اجرای ECP تحت تاثیر قرار می گیرند. در این مورد کنترل کننده مصرف CPU کاهش می یابد، چون که گره جدیدی به حلقه اضافه می شود و تلاش زیادی برای انتقال داده ها به گره جدید انجام

می شود سپس به این دلیل افزایش پیدا می کند که پس از کاهش و ثبات کم، پیکربندی مجدد در کنترل کننده نیز لازم است. بنابراین حتی در شرایط حجم کاری تصادفی، ADVISEمی تواند بینش مفیدی ارائه دهد که چگونه بخش های مختلف سرویس رفتار می کنند هنگامیکه ECPهای را اجرا می کنیم که برای دیگر بخش های سرویس قرار گرفته اند.

	ECP	انحراف معیار	زمان (s) متوسط ECP
سرویس ویدیو	ECP1	0	65
	ECP2	0	15
	ECP3	0	25
	ECP4	1.414	150
سرویس M2M	ECP1	4.5	45
	ECP2	1.4	20
	ECP3	0	20
	ECP4	1	75

Cloud Service	Observed Cloud Service Part	Elasticity Control Process	Average Standard Deviation	Maximum Variance	Minimum Variance
Video Service	Video Service	ECP_3	0.23	0.09	0.03
		ECP_4	0.61	0.99	0.23
	Distributed Video Storage Backend	ECP_3	0.28	0.14	0.034
		ECP_4	0.2	0.042	0.04
	Application Server	ECP_1	0.43	0.4	0.06
		ECP_2	0.31	0.47	0.01
M2M Service	Cloud Service	ECP_5	0.9	6.65	0.24
	Data End Service Topology	ECP_5	0.23	0.35	7.44E-05
	Event Processing Service Topology	ECP_7	1.1	4.9	0.046
		ECP_8	0.76	2.46	0.027
	Data Controller Service Unit	ECP_6	0.12	0.25	0
		ECP_8	0.22	0.41	0
	Data Node Service Unit	ECP_5	0.572	0.68	0.32
		ECP_6	0.573	1.4	0.07
	Event Processing Service Unit	ECP_7	1.08	3.59	0.11
		ECP_8	0.77	1.9	0.14

Table 4: ECPs effect estimation quality statistics

4.3 تاثیر موقتی ECP

جدول 3 زمان متوسط لازم برای تکمیل ECP را نشان می دهد. این اطلاعاتی که مخصوص برنامه است از اهمیت بالایی برخوردار می باشد و بر فرایند تصمیم گیری کنترل کننده کشش تأثیر می گذارد؛ زیرا نشان دهنده ی دوره

مهلت است که باید تا زمانیکه اثرات عملیات تغییر اندازه قابل توجه شوند منتظر باشد. به این ترتیب این اطلاعات، دانه بندی زمان را مشخص می کند که عملیات تغییر اندازه باید مورد توجه قرار گیرد. به عنوان مثال، متوجه می شویم که فرایند اضافه کردن و پیکربندی یک نمونه جدید برای عقبه ذخیره سازی خدمات ویدیویی نیاز به یک فاصله زمانی متوسط 150 ثانیه دارد که عمدتا زمان لازم برای دریافت و ذخیره داده ها از دیگر گره های حلقه است. اگر تصمیم گیری در فواصل کوتاهتر انجام شود، اثراتی که از عمل قبلی است بخشی از فرایند تصمیم گیری کنونی نخواهد شد.

4.4 کیفیت نتایج

ADVISEقادر است رفتار کشش بخش های مختلف سرویس را با در نظر گرفتن همبستگی بین متریک ها و ECP هایی که اجرا نشده اند برآورد کند. برای اینکه کیفیت نتایج خود را ارزیابی کنیم این واقعیت را در نظر گرفته ایم که ابزارهای موجود برآوردهایی با پیوستگی زمانی ایجاد نمی کنند. بنابراین، انتخاب می کنیم تا ADVISE را با محاسبه واریانس Var و انحراف معیار StdDev (معادله 9) با بیش از 100 برآورد ارزیابی کنیم که نتیجه کمی متفاوت است.

$$Var_{metric_i} = \frac{\sum (estMetric_i - obsMetric_i)^2}{nbEstimations - 1}, StdDev_{metric_i} = \sqrt{Var_{metric_i}} (9)$$

جدول 4 دقت نتایج ما را نشان می دهد. هنگامیکه دو سرویس را برابر می کنیم خدمات ویدئویی به دقت بالاتری دست پیدا می کند (انحراف معیار کوچکتر)، زیرا حجم بار اعمال شده بطور قابل ملاحظه پایدار است. با تمرکز بر دقت برآورد DaaS M2M مشاهده می کنیم که بر دانه بندی که این برآورد محاسبه شده و برECP بستگی دارد. علاوه بر این، انحراف معیار به متریک هایی که در بخش های مختلف سرویس ابری نظارت شده بستگی دارد. به عنوان مثال، در مورد سرویس ماشین به ماشین تعداد متریک مشتریان را می توان به سختی پیش بینی کرد چون که حسگرها اطلاعات خطا و یا مرتبط با هشدار را ارسال می کنند. جاییکه حداکثر واریانس برای تعداد مشتریان 4.9 است برای توپولوژی سرویس پردازش رخداد مشهود است.

به طور کلی، حتی در موقعیت های تصادفی بار سرویس ابری، چارچوب ADVISE اطلاعات دقیقی برای کنترل کننده های کشش تجزیه و تحلیل می کند و ارائه می دهد و به آنها اجازه می دهد تا کیفیت تصمیمات کنترل را با توجه به سیر تکامل متریک های نظارت شده در سطوح مختلف سرویس ابری بهبود بخشند. بدون استفاده این نوع برآورد، کنترل کننده های کشش باید از اطلاعات نمایه سازی سطح VM استفاده کنند، از این رو آنها باید سرویس های پیچیده ابری را کنترل کنند. این اطلاعات که از هر بخش سرویس است برای کنترل کشش سرویس های ابری پیچیده که مکانیسم های کنترل پیچیده را نشان می دهد ارزشمند است.

5.کار مربوطه

Verma وهمکاران (3) تأثیر عملیات پیکربندی مجدد بر روی عملکرد سیستم را مطالعه کردند. Verma وهمکاران عملیات پیکربندی مجدد سطوح زیرساخت را با عملیات برروی مهاجرت زنده متوجه شدند و مشاهده کردند که مهاجرت زنده VM تحت تاثیر استفاده CPU از ماشین مجازی منبع، هم از لحاظ مدت زمان مهاجرت و هم عملکرد برنامه است. نویسندگان با لیستی از توصیه های مربوط به تخصیص منابع پویا نتیجه گیری می کنند. Kaviani و همکاران (10) نمایه سازی را به گونه یک سرویس پیشنهاد می کنند که به سایر مشتری های ابری ارائه می شود و در تلاش برای پیدا کردن مصالحه بین دقت نمایه سازی، سرجمع عملکرد، و هزینه های متحمل شده است. Zhang و همکاران (4) الگوریتم هایی را برای ردیابی عملکرد برنامه های ابری پویا پیشنهاد می کنند و مقادیر متریک هایی مانند خروجی یا زمان پاسخ را پیش بینی می کنند. Shen و همکاران (5) چارچوب CloudScale را پیشنهاد می کنند که آن از پیش بینی منابع برای تخصیص خودکار منابع بر طبق اهداف سطح سرویس (SLO) با حداقل هزینه استفاده می کند. براساس پیش بینی تخصیص منابع، CloudScaleاز مهاجرت پیش بینی شده برای حل ناسازگاری های مقیاس گذاری (به عنوان مثال منابع کافی برای تطابق با نیازهای افزایش مقیاس وجود ندارد) و فرکانس و ولتاژ CPUبرای صرفه جویی در مصرف انرژی با حداقل تاثیرSLOs استفاده می کند. در مقایسه با این کار تحقیقاتی، با توجه به سطوح مختلف متریک ها، بسته به ساختار کاربردی که رفتار آن آموخته می شود، مدل خود را ساختیم.

علاوه بر این، عوامل تنش زای ملاحظه شده نیز به ساختار کاربردی و قابلیت های کشش (یعنی انواع عملیات) که برای نوع کاربرد فعال شده سازگار است. Juve و همکاران (11) سیستمی را پیشنهاد می کنند که در خودکاری تامین فرایند کاربردهای مبتنی بر ابر کمک می کند. Juve و همکاران دو مدل کاربردی، یکی کاربرد جریان کاری و یکی مورد ذخیره سازی داده را در نظر می گیرند و نشان می دهند که چگونه برای این موارد می توان برنامه های کاربردی را به طور خودکار مستقر و پیکربندی کرد. Li و همکاران (12) چارچوبCloudProfect را پیشنهاد می کنند که از رویدادهای منابع و وابستگی ها در میان آنها برای پیش بینی عملکرد برنامه های وب در ابر استفاده می کند.

در مقایسه با کار تحقیقاتی ارائه شده نه تنها بر برآورد تاثیر یک فرآیند کنترل کشش بر بخش سرویسی که با آن مرتبط است، بلکه بر بخش های مختلف سرویس ابری تمرکز می کنیم. علاوه بر این، رفتار کشش بخش های سرویس ابری مختلف را برآورد و ارزیابی می کنیم، زیرا نه تنها به تاثیر در پس از یک دوره مقرر تمایل داریم، بلکه به الگوی تاثیری که ECP مربوطه آن را معرفی می کند نیز متمایل هستیم.

6. نتیجه گیری و کار آینده

چارچوب ADVISEرا ارائه دادیم که قادر است رفتار بخش های سرویس ابری را برآورد کند هنگامیکه ECPهای مختلف را با توجه به انواع مختلف اطلاعات که از طریق نمودار وابستگی کشش ارائه شده اجرا می کنیم. بر اساس نتایج حاصل از دو سرویس ابری مختلف، نشان می دهیم که چارچوب ADVISE در واقع می تواند کنترل کننده های کششADVISE را در مورد رفتار سرویس ابری ارائه کند، که در جهت بهبود کشش سرویس ابری کمک می کند.

در کار آتی، قصد داریم ADVISE را با کنترل کننده کشش rSYBL ادغام کنیم [8] و مکانیزم های تصمیم جدیدی را ایجاد کنیم که اثرات پیوسته ECP را به عنوان ورودی ها در نظر بگیرند و تصمیمات را بر اساس رفتار مورد انتظار از هر بخش سرویس ارائه دهند.

References

1. Al-Shishtawy, A., Vlassov, V.: Elastman: Autonomic elasticity manager for cloudbased key-value stores. In: Proceedings of the 22Nd International Symposium on High-performance Parallel and Distributed Computing. HPDC '13, New York, NY, USA, ACM (2013) 115–116

2. Wang, W., Li, B., Liang, B.: To reserve or not to reserve: Optimal online multiinstance acquisition in iaas clouds. In: Presented as part of the 10th International Conference on Autonomic Computing, Berkeley, CA, USENIX (2013) 13–22

3. Verma, A., Kumar, G., Koller, R.: The cost of reconfiguration in a cloud. In: Proceedings of the 11th International Middleware Conference Industrial Track. Middleware Industrial Track '10, New York, NY, USA, ACM (2010) 11–16

4. Zhang, L., Meng, X., Meng, S., Tan, J.: K-scope: Online performance tracking for dynamic cloud applications. In: Presented as part of the 10th International Conference on Autonomic Computing, Berkeley, CA, USENIX (2013) 29–32

5. Shen, Z., Subbiah, S., Gu, X., Wilkes, J.: Cloudscale: elastic resource scaling for multi-tenant cloud systems. In: Proceedings of the 2nd ACM Symposium on Cloud Computing. SOCC '11, New York, NY, USA, ACM (2011) 5:1–5:14

6. Moldovan, D., Copil, G., Truong, H.L., Dustdar, S.: Mela: Monitoring and analyzing elasticity of cloud services. In: 2013 IEEE Fifth International Conference on Cloud Computing Technology and Science (CloudCom). (2013) 7. OASIS Committee Specification Draft 01: Topology and Orchestration Specification for Cloud Applications Version 1.0. (2012) 8. Copil, G., Moldovan, D., Truong, H.L., Dustdar, S.: Multi-level Elasticity Control of Cloud Services. In Basu, S., Pautasso, C., Zhang, L., Fu, X., eds.: ServiceOriented Computing. Lecture Notes in Computer Science. Springer Heidelberg 9. Trihinas, D., Pallis, G., Dikaiakos, M.D.: JCatascopia: Monitoring Elastically Adaptive Applications in the Cloud. In: 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. (2014) 10. Kaviani, N., Wohlstadter, E., Lea, R.: Profiling-as-a-service: Adaptive scalable resource profiling for the cloud in the cloud. In Kappel, G., Maamar, Z., MotahariNezhad, H., eds.: Service-Oriented Computing. Volume 7084 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 157–171 11. Juve, G., Deelman, E.: Automating application deployment in infrastructure clouds. In: Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science. CLOUDCOM '11, Washington, DC, USA, IEEE Computer Society (2011) 658–665 12. Li, A., Zong, X., Kandula, S., Yang, X., Zhang, M.: Cloudprophet: towards application performance prediction in cloud. In: Proceedings of the ACM SIGCOMM 2011 conference. SIGCOMM '11, New York, NY, USA, ACM (2011).