قابلیت ارتجاعی ابر:

بررسی

چکیده

قابلیت ارتجاعی ابر ویژگی منحصر به فرد محیطهای ابری است که تأمین (کاهش تأمین) تقاضا و یا تنظیم مجدد منابع توسعه ابر را می‌پذیرد. مدیریت کارایی قابلیت ارتجاعی ابر چالشی است که توجه جامعه پژوهشی را به خود جلب کرده است. این کار به منزله بررسی تلاش‌های پژوهشی در این راستا است. سهم اصلی این کار، مرور دقیق آخرین روش‌های مدیریت ارتجاعی و ارائه طرح دقیق طبقه‌بندی با تمرکز بر روش‌های تصمیم‌گیری ارتجاعی است. در پایان، درباره چالش‌های تحقیقاتی مختلف و جهت‌های تحقیقاتی بیشتر درباره کلیه مراحل قابلیت ارتجاعی بحث می‌کنیم که می‌توان به عنوان حالت خاص رفتار ارادی سیستم‌های محاسباتی تلقی کرد (این تحقیق با همکاری اتحادیه اروپا (صندوق اجتماعی اروپا- ESF) و صندوق‌های ملی یونان از طریق برنامه عملیاتی «آموزش و یادگیری بلند مدت چارچوب مرجع استراتژیک ملی (NSRF)- برنامه سرمایه‌گذاری تحقیقاتی: تالس[1]. سرمایه‌گذاری در جامعه دانش از طریق صندوق اجتماعی اروپا» انجام شده است).

1. مقدمه

محاسبات ابر مدل توسعه‌ای را ارائه می‌دهد که هدفش کاهش هزینه‌های لحظه‌ای منابع محاسباتی از طریق تنظیم اجاره منابع مجازی پویا است که می‌تواند بر پایه تقاضا تشکیل شود. منابع مجازی نسخه‌های مجازی دنیای واقعی

[1] . Thales

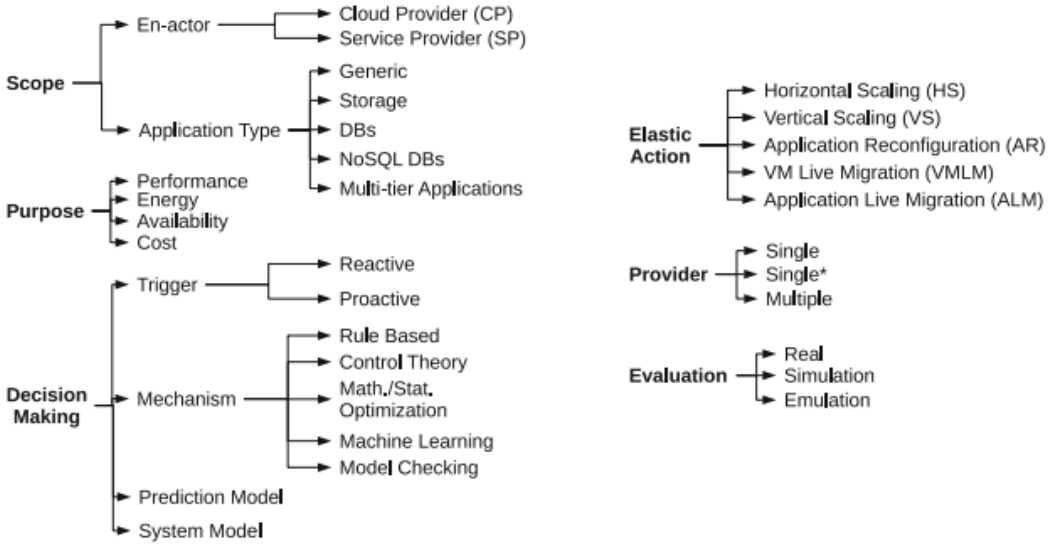
هستند، اغلب به صورت ماشین‌های مجازی (VMs) هستند که از تکنولوژی‌های مجازی‌سازی استفاده می‌کنند [65]. مدل پیشنهادی پرداخت بهای خرید با مدیریت منابع انعطاف‌پذیر به پذیرش وسیع در زمینه استفاده از ابر کمک کرده است، زیرا مشتری موظف است تنها برای منابع مورد استفاده بهایی پرداخت کند. بدین ترتیب، محاسبات ابری نه تنها قادر به ارائه منابع محاسباتی از راه دور (به عنوان مثال، ماشین‌های مجازی) گزینه‌های اصلی برای مؤسسات علمی هستند بلکه قادر به ارائه منابع در هر اندازه‌ای از سازمان‌ها و شرکت‌ها هستند. با این وجود، مدیریت منابع کارا جنبه کلیدی برای کاهش هزینه توسعه است.

کارهای متعددی وجود دارند که روش‌های مختلف مکانیزم ارتجاعی را ارائه می‌دهند. در این مقاله تمرکزمان بر تمامی جنبه‌های ارتجاعی است اما به طور خاص قصد داریم که در رابطه با مدل‌های پایه‌ای، مکانیزم تصمیم‌گیری را مورد استفاده قرار دهیم. علاوه بر این، قصد داریم از طریق طبقه‌بندی‌مان روش‌های مختلفی را توسعه داده و مقایسه کنیم که امروزه تمایل به توسعه در انزوا دارند.

روش‌های ارتجاعی را به عنوان موضوعات بین رشته‌ای دو حوزه اصلی رایانه‌ای توزیع شده/ ابر و محاسبات ارادی می‌سنجیم. همانند حوزه محاسبات ارادی، شامل 4 مرحله حلقه MAPE است [44]، یعنی، نظارت، تحلیل، برنامه‌ریزی و اجرا. هر مرحله متمایز، چالش‌های تحقیقی منحصر به فردی را ارائه می‌دهد که در آثار ارائه شده با روش‌های مختلف مورد توجه قرار گرفته است. در این مقاله، بیشتر بر سه مرحله آخر متمرکز می‌شویم.

برخی از تلاش‌ها در گذشته برای ایجاد دید کلی در حوزه ارتجاعی صورت گرفته است، به عنوان مثال، [26] مکمل کار ماست، اما بیشتر بر ابزارها، معیارها و حجم کار متمرکز است. راهبردهای ارتجاعی را به طور گسترده‌تر به عنوان مکانیزم تصمیم‌گیری ارتجاعی شرح می‌دهیم. [48] نیز مکمل کار ماست، اما پیشنهادات جدیدتری ارائه می‌دهیم و طیف گسترده‌ای از اقدامات و اهداف ارتجاعی را پوشش می‌دهیم. بررسی قدیمی‌تر و محدودتری در [33] انجام شده است. بررسی کلی سیستماتیک درباره خدمات تجاری ابر در [46] انجام شده است، که در آن نویسندگان چالش اصلی درباره ویژگی ارتجاعی را مطرح کرده‌اند. همچنین، کار ما شکاف مهم درباره این مسأله را پر می‌کند.

ساختار این مقاله بررسی به شرح زیر است. در بخش 2، دسته‌بندی و جدول طبقه‌بندی را ارائه می‌دهیم. در بخش 3، جزئیات بیشتری درباره هر بعد از طبقه‌بندی‌مان ارائه می‌دهیم و یافته‌های اصلی را به تصویر می‌کشیم. در فصل 4 نتیجه‌گیری را ارائه می‌دهیم.



شکل 1. طرح دسته‌بندی

2. دسته‌بندی و طبقه‌بندی

به منظور ارائه طبقه‌بندی مختصر روش‌های موجود برای ارتجاع ابر، ابتدا طبقه‌بندی ارائه می‌دهیم که ما را قادر می‌سازد تا جنبه‌های متمایز پیشنهادات مختلف را روشن کنیم. طبقه‌بندی در شکل 1 خلاصه شده و شامل ابعاد زیر است:

- محدوده. این جنبه به دو دسته طبقه‌بندی می‌شود: (1) برقرار کننده و (2) نوع کاربرد. اولی نشان می‌دهد که آیا روش ارتجاعی توسط ارائه دهنده زیرساخت ابر (ارائه دهنده ابر (CP)) اعمال می‌شود یا توسط کاربر زیرساخت ابر اعمال می‌شود که مدیریت برنامه‌های ابر را در بالای زیرساخت ابر (ارائه دهنده سرویس (SP)) اعمال می‌کند. نوع کاربرد نشان می‌دهد که پیشنهاد به مدیریت ارتجاعی نوع خاصی از کاربر ابر در لیست زیر اشاره دارد: پایگاده

داده‌های ارتباطی (DBs)، پایگاه داده‌های NoSQL (NoSQL DBs)، برنامه‌های چندلایه (به عنوان مثال، برنامه‌های کاربردی وب خاص)، Generic (در صورتی که ابزار برنامه کاربردی- غیرارادی) و یا ذخیره‌سازی.

- هدف. در این بعد، روش‌ها را طبق هدف اقدامات ارتجاعی دسته‌بندی می‌کنیم. هدف می‌تواند یکی از موارد زیر باشد: (1) عملکرد، (2) دسترسی، (3) هزینه، (4) انرژی. عملکرد مربوط به تعمیر و حفظ یا تضمین قابل قبول کاربرد و یا توافق سطح خدمات (SLA) مشخص شده عملکرد نرم‌افزار است. در دسترس بودن به درجه‌ای اشاره دارد که در آن برنامه‌ها و منابع در زمان قابل اجرا بوده و در زمان مورد نیاز کاربر نهایی قابل اعتماد باشند [42]. هزینه به کاهش هزینه‌های عملیاتی برنامه‌های توسعه‌یافته در ابر اشاره دارد که عموماً هدف عملکرد را نیز حفظ می‌کنند و یا آستانه هزینه‌ها را در محدودیت‌های عملکرد خاص نگه می‌دارند. در پایان، طبقه‌بندی انرژی، به طور دقیق مربوط به هزینه است اما شامل روش‌های ارتجاعی نیز می‌باشد که به طور مستقیم در به حداقل رساندن میزان انرژی کمک می‌کنند.

- تصمیم‌گیری. چهار معیار دسته‌بندی متمایز وجود دارد که روش تصمیم‌گیری هر کار را در طبقه‌بندی ما مشخص می‌کند، یعنی (1) قابلیت اطمینان، که نشان دهنده آن است که آیا مکانیزیم ارتجاعی در روش واکنش‌پذیر یا پیشگیرانه ایجاد می‌شود یا خیر؛ (2) مکانیسم که به روش تصمیم‌گیری اشاره دارد؛ (3) مدل پیش‌بینی (PM) که نشان دهنده استفاده از مدلی برای پیش‌بینی تغییرات بار ورودی آینده و یا ارزش‌های ارزیابی خاص است؛ و (4) مدل سیستم (SM) که به استفاده از مدل برای نشان دادن رفتار (ارتجاعی) سیستم اشاره دارد که در آن سیستم به طور کاملاً ارتجاعی (مانند، صف) ساخته شده است. مکانیزم‌های انعطاف‌پذیر بیشتر به دسته‌های زیر طبقه‌بندی می‌شوند: (1) مبتنی بر قانون، (2) بهینه‌سازی ریاضی/ آماری، (3) یادگیری ماشین، (4) نظریه کنترل و (5) بررسی مدل با توجه به زمینه اصلی که متعلق به سیاست ارتجاعی است.

- عمل ارتجاعی. قابلیت ارتجاعی منبع ابر را می‌توان در فرم‌های مختلف اعمال کرد و مربوط به تغییرات در (1) اندازه (مقیاس عمودی (VS))، (2) مکان (تغییر مکان در ماشین‌های مجازی (VMLM)) یا (3) تعداد ماشین‌های مجازی به کار رفته (مقیاس افقی (HS)) است. مثال‌هایی از این نوع قابلیت ارتجاع به ترتیب عبارتند از تخصیص

حافظه بیشتر یا CPU برای ماشین‌های مجازی، حرکت ماشین‌های مجازی به سوی ماشین فیزیکی کمتر بارگذاری شده و افزایش تعداد ماشین‌های مجازی خوشه برنامه. علاوه بر این، عمل ارتجاعی شامل دو نوع ارتجاع دیگر است، (4) پیکربندی مجدد برنامه (AR)، که در آن ابزار الاستیک قادر به تغییر جبنه‌های کاربردی خاص هستند (به عنوان مثال، اندازه دریافت پایگاده داده‌های ارتباطی) و (5) تغییر مکان برنامه کاربردی (ALM) که در آن اجزای کاربرد خاص در ماشین‌های مجازی مانند پایگاده داده تغییر مکان می‌دهند.

- ارائه دهنده. این طبقه‌بندی به دسته‌بندی تعداد ارائه‌دهندگان زیرساخت ابر اشاره دارد که به طور همزمان از ابزار ارتجاعی پشتیبانی می‌کنند. مقادیر احتمالی (1) منفرد، که تنها نشان‌دهنده پشتیبانی از ارائه دهنده ابر است، (2) منفرد*، که نشان‌دهنده پشتیبانی از بیش از یک ارائه دهنده است، با وجود آن که همزمان نیستند، و (3) چندگانه، که در آن کنترل ارتجاع ارائه‌دهندگان ابر چندگانه را به طور همزمان گسترش می‌دهد.

- ارزیابی. در پایان، آخرین جنبه به نوع ارزیابی هر کار اشاره دارد. مقادیر احتمالی عبارتند از: (1) شبیه‌سازی، که در آن نتایج بر پایه محاسبات در محیط مصنوعی شبیه‌سازی‌شده بدست می‌آید (به عنوان مثال، OMNeT++)، (2) واقعی، که در آن ابزار ارتجاعی بر پایه زیرساخت حقیقی ابر مورد استفاده قرار می‌گیرند.

جدول 1. طبقه‌بندی پیشنهادات پژوهشی با توجه به دسته‌بندی ما

مرجع	محدوده		هدف	تصمیم‌گیری				عمل ارتجاعی	ارائه دهنده	ارزیابی
	برقرارکننده	نوع کاربرد		حرکت	مکانیسم	PM	SM			
[54]	CP	عمومی	Perf.	پیشگیرانه	بهینه‌سازی ریاضی/ آماری	x	x	HS VMLM	Single	Real
[21]	CP	Generic	Perf.	واکنش‌پذیر	حقیقی بر پایه			HS	Single[a]	Real
[37]	CP	Generic	Perf.	Reactive	Rule based			HS VS	Single	Real
[50]	CP	Generic	Perf. energy	Reactive	Math./stat. optimization			VMLM	Single	Real
[61]	CP	Generic	Perf. energy	Reactive proactive	Rule based math./stat. optimization	x		VS VMLM	Single	Real
[68]	CP	برنامه‌های چندلایه	Perf.	Proactive	Mach. learn.		x	HS VS	Single	Real
[31]	SP	DBs	Perf.	Reactive	Mach. learn.		x	ALM	Single	Real
[20]	SP	DBs	Perf.	Reactive proactive	Rule based	x		HS VMLM ALM	Single	Real
[59]	SP	DBs	Perf.	Proactive	Rule based math./stat. optimization			HS ALM	Single	Real
[55]	SP	Generic	Avail.	Reactive	Rule based			HS	Multiple	Real
[63]	SP	Generic	Perf.	Proactive	Mach. learn.	x	x	VS VMLM	Single	Real
[39]	SP	Generic	Perf.	Proactive	Rule based math./stat. optimization	x	x	HS	Single	Real
[60]	SP	Generic	Perf.	Reactive	تئوری کنترل	x	x	HS	Single[a]	Real
[49]	SP	Generic	Perf.	Reactive	Rule based		x	HS	Single[a]	Real
[17]	SP	Generic	Perf.	Reactive proactive	Rule based	x		HS	Single	Simulation
[16]	SP	Generic	Perf.	Reactive proactive	Control theory		x	HS	Single	Simulation
[51,52]	SP	Generic	Perf.	Reactive proactive	Rule based mach. learn.	x	x	HS	Single	Simulation
[29]	SP	Generic	Perf. cost	Proactive	Mach. learn. math./stat. optimization	x	x	HS VS	Single	Real
[25]	SP	Generic	Perf. cost	Proactive	Rule based		x	HS AR	Multiple	Real
[66]	SP	Generic	Perf. cost	Reactive	Rule based		x	HS VS AR	Single	Real
[40]	SP	Multi-tier applications	Perf.	Proactive	Control theory			VS	Single	Real

(*Continued*)

Table 1. (Continued)

Citation	Scope		Purpose	Decision making				Elastic action	Provider[a]	Evaluation
	Enactor	Appl. type		Trigger	Mechanism	PM	SM			
[43]	SP	Multi-tier applications	Perf.	Reactive proactive	Rule based math./stat. optimization		x	HS	Single	Real
[38]	SP	Multi-tier applications	Perf.	Reactive proactive	Rule based math./stat. optimization		x	HS	Single	Real
[35]	SP	Multi-tier applications	Perf. avail.	Reactive	Rule based		x	HS	Single	Real
[56]	SP	Multi-tier applications	Perf. avail.	Reactive	Rule based			HS VS	Multiple	Real
[32]	SP	Multi-tier applications	Perf. cost	Proactive	Math./stat. optimization	x	x	HS	Single	Real
[22]	SP	Multi-tier applications	Perf. cost	Proactive	Math./stat. optimization		x	HS AR	Single	Simulation
[18]	SP	Multi-tier applications	Perf. cost	Proactive	Rule based control theory	x		HS AR	Single	Real
[36]	SP	Multi-tier applications	Perf. cost	Reactive	Rule based math./stat. optimization		x	HS	Single	Real
[57]	SP	Multi-tier applications	Perf. cost	Reactive	Rule based math./stat. optimization			HS	Single	Emulation
[62]	SP	Multi-tier applications	Perf. cost	Reactive proactive	Rule based	x		VS	Single	Real
[41,45,64]	SP	NoSQL DBs	Perf.	Proactive	Mach. learn.		x	HS	Single[a]	Real
[58]	SP	NoSQL DBs	Perf.	Proactive	Mach. learn. math./stat. optimization		x	HS AR	Single	Real
[53]	SP	NoSQL DBs	Perf.	Proactive	Model checking mach. learn. math./stat. optimization		x	HS	Single[a]	Emulation
[15]	SP	NoSQL DBs	Perf.	Reactive	Control theory mach. learn.		x	HS	Single	Real
[23]	SP	NoSQL DBs	Perf.	Reactive	Rule based			HS AR	Single	Real
[47]	SP	NoSQL DBs	Perf.	Reactive	Rule based control theory			HS	Single	Real
[27]	SP	NoSQL DBs	Perf.	Reactive	Rule based math./stat. optimization			HS AR	Single	Real
[19]	SP	Storage	Perf.	Reactive proactive	Rule based			HS	Single	Real

[a]Multiple providers supported, not simultaneously

بر پایه طبقه‌بندی بالا، پیشنهادات موجود برای ارتجاع ابر در جدول 1 ارائه شده است. طبقه‌بندی بالا نوع اطلاعات بازخورد جمع‌آوری‌شده توسط محیط را برای اجرای تصمیم‌گیری و اجرای ارتجاع پوشش نمی‌دهند، زیرا به نظر می‌رسد نوع بازخورد نقش کمتری در طبقه‌بندی پیشنهادات بازی می‌کند. به طور خاص، تمام پیشنهادات برای کمک به تصمیم‌گیری از یک مکانیزم برای نظارت بر معیارهای خاص سیستم/ شبکه/ نرم‌افزار استفاده می‌کنند. برای مقابله با خوشه بار احتمالی یا ناپایداری‌های ارزیابی، از بسیاری از آثار روش‌های هموارسازی مانند میانگین حرکت وزن دار نمایی (EWMA)، میانگین حرکت نمایی (EMA) و یا میانگین حرکت (MA)، استفاده می‌شود. جزئیات بیشتر به علت محدودیت فضا حذف شده است.

3. خلاصه روش‌های موجود

در این بخش، با توجه به روش‌های اصلی، روشی را با جزئیات برای هر بعد طبقه‌بندی ارائه می‌دهیم.

3.1. محدوده

اولین جنبه بعد محدوده نشان می‌دهد که چه کسی مسئول مکانیسم ارتجاعی است. در چنین طرحی، روش ارتجاعی همراه با زیرساخت ابر هسته است و روش‌های مربوطه به عنوان ابر ارائه دهنده خاص تعریف شده‌اند. به عنوان مثال، [37] بر پایه ابزاری است که در بالای زیرساخت‌های IaaS نصب شده است، سیستم DejaVu در [68] قابلیت این زیرساخت‌ها را گسترش داد. مجموعه دیگر پیشنهادات به امتیاز خاصی برای منابع (به عنوان مثال، [54] به ماژول KVM سفارشی و رابط وابسته است، [21] در OpenStack تجمیع شده است و [61] ولتاژ و فرکانس CPU را تنظیم می‌کند) و دسترسی به اطلاعاتی نیاز دارند که تنها ارائه‌دهنده ابر قادر به ارائه آن است (به عنوان مثال، [50] به اطلاعات محلی دستگاه فیزیکی وابسته است). با این وجود، بیشتر پیشنهادات، ارتجاع پیشرفته‌ای را برای خدمات مبتنی بر ابر بدون در نظر گرفتن تعمیم ویژگی‌های ارتجاعی خدمات ابر ارائه می‌دهند؛ آنهایی که در ارتباط با ارائه‌دهندگان خدمات تعیین می‌شوند.

با توجه به نوع کاربردی که پیشنهادات بر آن متمرکز شده‌اند، اکثر آنها مستقل از برنامه کاربردی چندمنظوره هستند و یا برای برنامه‌های کاربردی چندمنظوره مبتنی بر وب طراحی شده‌اند. اکثر پیشنهادها از مدیریت ارتجاعی تمام سه سطح برنامه وب (یعنی، وب سرور، سرور برنامه، سرورهای ذخیره‌سازی) پشتیبانی می‌کنند، به جز در [18 و 32] که تنها قابلیت ارتجاعی سطح سرور برنامه را مدیریت می‌کنند، و [62 و 68] که به سادگی خدمات وب را مورد هدف قرار می‌دهند. بخش قابل توجهی از پیشنهادات ارتجاعی، ناحیه NoSQL را مورد هدف قرار می‌دهند. روش‌ها در این گروه سیستم- خاص (به عنوان مثال، [23] اهداف Cassandra، [27] هدف HBase، [58] اهداف بی‌انتها، در حالی که [47] HDFS را در نظر می‌گیرد) و یا قابل اجرا برای مجموعه بزرگی از سیستم‌های NoSQL مانند Cassandra، HBase، Voldemort و Infinispan [15، 41، 45، 53 و 64] هستند. در [20، 31 و 59] انعطاف‌پذیری در پایگاه‌های داده ارتباطی در نظر گرفته می‌شود. [20 و 31] را می‌توان در هر پایگاه داده ارتباطی مورد استفاده قرار داد زیرا موتور پایگاه داده را اصلاح نمی‌کنند، در حالی که در [59] موتور پایگاه داده برای پشتیبانی از حرکت‌های زنده با الهام از روش [30] اصلاح می‌شود. در پایان روش منفردی وجود دارد که به عنوان Storage دسته‌بندی می‌شود [19]، که در آن قابلیت ذخیره‌سازی ارتجاعی عملکرد ذخیره‌سازی ماشین مجازی از طریق روش‌های ذخیره‌سازی در نظر گرفته می‌شود.

3.2. هدف

تمامی روش‌های موجود در جدول 1 به منظور بهبود عملکرد می‌باشد. تنها [55] استثنا است که در آن هدف ارتجا دستیابی را از طریق استفاده از ارائه‌دهندگان چندابری افزایش می‌دهد. هدف عملکرد می‌تواند ثابت باشد (به عنوان مثال، در توافق سطح خدمات یا به عنوان آستانه تعریف شده توسط کاربر تعریف شود) و یا به عنوان ناظر مستمر و بهینه‌سازی ابزار سیستم بیان شود. در چنین کارهایی، کاهش هزینه‌های مالی به طور غیرمستقیم از طریق استفاده از منابع کمی در نظر گرفته می‌شود که عملکرد را به طور قابل قبولی حفظ می‌کنند. در [35 و 36]، هدف هملکرد با ارائه تضمین‌های دسترسی همراه است.

برخی از پیشنهادات به صراحت با هدف کاهش هزینه مالی بیان شده‌اند. به طور خاص در [29، 36 و 57] برآورد هزینه برای مقیاس در درون یا خارج ابر به کار گرفته شد، در حالی که در [25] از تخمین‌های مشابه برای انتخاب میان گسترش در زیرساخت ابر عمومی یا خصوصی استفاده می‌شود و در [62] تحلیل بازده سرمایه‌گذاری (ROI) پیش از گسترش حقیقی ارائه می‌شود. در سایر روش‌ها قابلیت ارتجاعی بنا بر محدودیت بودجه است. به طور خاص، [66] در صورتی که حداکثر هزینه بیش از هزینه قابل دسترس باشد، مانع مقیاس‌گذاری می‌شود، [22] اجرای تنظیم مجدد برنامه (یعنی، پاسخ‌های سرور متنی برای صرفه‌جویی در پهنای باند) را حفظ می‌کند تا هزینه‌ها پایین‌تر از حد بودجه باقی بماند و [32] دسته‌بندی بودجه (یعنی، دسته‌بندی فلزات: طلا/ نقره/ برنز) را ارائه می‌دهد که محدودیت‌های مقیاس‌گذاری منابع را تنظیم می‌کنند. در پایان، [18] تلاش می‌کند تا چندین برنامه کاربردی را در ماشین‌های مجازی هماهنگ کند تا هزینه تأمین شده را کاهش دهد.

علاوه بر این، دو مقاله وجود دارد که صرفه‌جویی در انرژی را با هدف عملکرد در نظر گرفته‌اند. در [50] حرکت زنده به کار گرفته می‌شود تا همانند بسیاری از ماشین‌ها در حالت خواب تنظیم شود، در حالی که در [61]، منابع ماشین‌های مجازی برای مقادیر ولتاژ و فرکانس پویا برای صرفه‌جویی در انرژی مورد استفاده قرار می‌گیرند.

3.3. تصمیم‌گیری

راه‌اندازی فرایند تصمیم‌گیری. مقالات به (1) واکنشی، (2) پیشگیرانه و (3) ترکیب واکنش‌پذیر و پیشگیرانه تقسیم می‌شوند. از یک طرف، روش‌های واکنشی معمولاً بر پایه پیگیری مستمر معیارهای ویژه و اعتبارسنجی قوانین مبتنی بر آستانه استوار هستند. فرایند تصمیم‌گیری اغلب با تخلف از آستانه منفرد به وجود می‌آید. با این وجود، فرایند تصمیم‌گیری تنها پس از یک دوره مشخص می‌تواند آستانه را نقض کند (به عنوان مثال، [55 و 57]) و یا تعداد ارزیابی‌های نقض شده را برای پرهیز از واکنش بیش از حد از قبل تعیین کند (به عنوان مثال، [21]). از سوی دیگر، روش‌های پیشگیرانه مکانیزمی هستند که برای پیش‌بینی تنوع بار آینده و یا رفتار آتی سیستم مورد استفاده قرار می‌گیرند. با این وجود، روش‌های صرفاً پیشگیرانه از این واقعیت رنج می‌برند که قادر به مقابله با افزایش ناگهانی

بار کاری نیستند. برای غلبه بر این نگرانی، در مقالاتی مانند [17، 20 و 43] روش ترکیبی در نظر گرفته شده است. علاوه بر این، در [16 و 38] پیشنهاد شد که از روش‌های واکنشی برای مقیاس‌بندی خارجی و از روش‌های پیشگیرانه برای مقیاس‌بندی داخلی استفاده شود. برای اولین بار از سازگاری سریع برای خوشه حجم کار استفاده شد. همچنین مقالاتی وجود دارند که از روش‌های واکنشی در حالتی استفاده می‌کنند که مکانیسم پیشگیرانه از تصمیم‌گیری نامطمئن است [61]، یا پیش‌بینی کننده به اندازه کافی برای تصمیم‌گیری مناسب آموزش ندیده است [51 و 52]. [62] ترکیبی از روش‌های واکنش‌پذیر و پیشگیرانه را ارائه داد که در ادامه بر پایه روزانه فعال می‌شوند. در پایان، در [19] روش‌های واکنش‌پذیر و پیشگیرانه‌ای ارائه شدند که به طور همزمان فعال نیستند اما می‌توان هر یک از آنها را به طور جداگانه پشتیبانی کرد.

مکانیسم تصمیم‌گیری. در بخش قبلی از روش‌های اصلی استفاده شده در تصمیم‌گیری ارتجاعی استفاده کردیم. در اینجا، این موضوع را دقیق‌تر توضیح می‌دهیم، ابتدا کاربرد روش‌های متداول و سپس روش‌های ترکیبی را شرح می‌دهیم.

با این وجود، روش‌های متناظر لزوماً ساده نیستند. به عنوان مثال، [17] از دسته مدل‌های پیش‌بینی به طور همزمان استفاده می‌کند تا برآورد بار و نقاط آستانه بالقوه آتی را بررسی کند. همچنین، [20] از مدل پیش‌بینی برای برآورد بار برای مقیاس پیشگیرانه بر پایه قوانین مشخص استفاده می‌کند. مدل‌سازی سیستم به طور کلی سیاست‌های تصمیم‌گیری را بهبود می‌بخشد. در [49]، سیستم به عنوان صفی از مشاغل مدل‌سازی شد و به اقدامات انعطاف‌پذیری در زمان ورود یا تکمیل شغل منتهی شد. دو روش مبتنی بر قواعد مبتنی بر مدل‌های سیستم غیربدیهی هستند [35]، که در آن سیستم به طور خودکار در حالت‌های مختلف به دلیل اجرای قانون مدل‌سازی می‌شود و در [25] حالتی در نظر گرفته شد که در آن مدل گراف، تأثیر قوانین ارتجاعی را در کل سیستم پذیرفته است. به عنوان مثال نهایی، در [39] روش مبتنی بر قاعده فازی ارائه شد که در آن کاربر قوانین را در فرمی مشخص می‌کند: «اگر بار کاری بالا باشد، و زمان پاسخ کند باشد، آنگاه دو ماشین مجازی بیشتر به منابع موجود افزوده

می‌شوند»، بدون نیاز به توصیف ارزش‌های «بالا» و «کند»؛ این ارزش‌ها بر پایه اطلاعات ارائه شده توسط ذی‌نفعان فنی محاسبه می‌شوند.

سیاست مبتنی بر بهینه‌سازی ریاضی/ آماری. این روش‌ها مسأله ارتجاعی را به عنوان بهینه‌سازی مدل‌سازی می‌کنند. در [32] روش مقیاس‌پذیری بهینه به پیروی از روش شاخه‌ای ارائه شد و پس از انجام تحلیل سری‌های پیچیده به پیش‌بینی بار اضافی آتی محدود شد. در [22]، تصمیم‌گیری واکنش‌پذیر مسأله حداکثر رساندن را در برنامه‌های دینامیکی کاهش می‌دهد؛ این روش از مدل صف به عنوان مرحله پیش‌پردازش برای تعیین مزایای بالقوه اشتغال انواع الگوریتم‌های مختلف برای خودسازگاری استفاده می‌کند. روش [50] به بررسی‌های برنولی کمک کرد تا ماشین مجازی مناسب را برای هدایت حرکت زنده پیدا کنند. در پایان، بهینه‌سازی به مدل‌سازی سیستمی اشاره می‌کند که پس از آن ارتجاع را در اختیار می‌گذارد؛ به عنوان مثال، [54] از پروفایل‌های آنلاین و منحنی مناسب برای تولید مدل عملکرد استفاده کرد که قادر به پیش‌بینی آن است که آیا برنامه در حالت تناقض با هدف است یا خیر.

سیاست مبتنی بر یادگیری ماشین. یادگیری ماشین معمولاً در تصمیم‌گیری ارتجاعی مورد استفاده قرار می‌گیرد. [68] مدل سیستم را در قالب طبقه‌بندی ارائه داد، در حالی‌که خوشه‌ها در گروه‌های نمایشی نیز کار می‌کنند. تصمیم‌گیری‌های ارتجاعی قبلی برای گروه مشابه در تصمیم‌گیری‌های آتی تأثیر می‌گذارند. روش [31] مشابه است. در [63] مدل پیش‌بینی مبتنی بر زنجیره مارکوف برآوردهایی را ارائه داد که به طبقه‌بندی چندمتغیره برای دسته‌بندی حالت‌های عادی یا غیرعادی آتی می‌پردازند و اقدامات ارتجاعی را مطابق با آن انجام می‌دهند. مثال روش‌های پیشرفته‌تر در [41، 45 و 64] ارائه شده است که در آنها روش Q-Learning از محاسبه ارزش‌های حالت بهینه به منظور حل غیرمستقیم مدل تصمیم‌گیری مارکوف (MDP) برای توصیف سیستم پیروی می‌کند.

سیاست‌های تئوری کنترل. تئوری کنترل حوزه علمی است که قادر به ارائه راه‌حل‌های محاسباتی ارادی است و با سیاست‌های ارتجاعی خاصی در نظر گرفته می‌شود. به عنوان مثال، [60] از روش تئوری کنترل استفاده کرد که در آن نمایشگر مدل‌سازی صف در سیستم ایجاد می‌شود و پیش‌بینی کننده را نیز به کار می‌گیرد. همچنین در [40]

نمونه از کاربرد فیلترهای کالمن[1] و کنترل بازخورد حاصل از ارتجاع ارائه شد. در پایان، در [16] روش کنترلی با ترکیب سیاست‌های فعال و واکنشی مورد بررسی قرار گرفت. همانند سایر مقالات مشابه؛ زیرساخت ابر به عنوان صف مدل‌سازی می‌شود، در حالی که برآوردکنندگانی برای بار اضافی خارجی در نظر گرفته می‌شود.

سیاست‌های ترکیبی. سیاست‌های ذکر شده مطابق با ماژول‌های تصمیم‌گیری است که یکی از مکانیسم‌های مشخص شده را استفاده می‌کنند. با این وجود، روش‌های ارتجاعی اغلب چند مکانیسم را به کار می‌گیرند، همان‌طور که در ادامه بررسی می‌شود.

شایع‌ترین روش ترکیبی، ترکیب قوانین با یکی از مکانیسم‌های آزاد است. قوانین می‌توانند به طور مؤثر راه‌حل‌های تئوری کنترل را گسترش دهند. بر مثال، در [47]، کنترل‌کننده یکپارچه‌ای ارائه شد که بر پایه آستانه مناسب به صورت پویا از آستانه برای تنظیم CPU بالا و پایین مورد استفاده در تصمیم‌گیری‌های ارتجاعی استفاده می‌کند. در [18]، رگرسیون خطی برای پیش‌بینی بار آتی مورد استفاده قرار گرفت و سپس مقادیر پیش‌بینی شده در مدل سازگار با متغیر وابسته به کنترل‌کننده متناسب آزاد مورد استفاده قرار گرفتند. تصمیمات نهایی درباره تعداد ماشین‌های مجازی که باید اضافه یا حذف شوند بر پایه سیاست مبتنی بر قانون اتخاذ می‌شوند. قوانین را می‌توان با روش‌های یادگیری ماشین ترکیب کرد. نمایش این دسته ترکیبی در [51 و 52] اراه شده است، که در آن سه مدل از ابزار WEKA برای حمایت از تصمیم‌گیری استفاده می‌کنند. مدل اول، پیش‌بینی کننده سری زمانی است که حجم کار آتی را پیش‌بینی می‌کند. مدل دوم، مدل بیز است که به طور قابل ملاحظه‌ای به روزرسانی می‌شود و رابطه میان بار کار فعلی و روش دسته‌بندی نقض آستانه را می‌آموزد، و مدل سوم نیز یک مدل نوآوری بیز است که تعداد مطلوب ماشین‌های مجازی را برآورد می‌کند.

خانواده دیگر راه‌حل‌های ترکیبی، ترکیب قوانین با برخی از انواع بهینه‌سازی‌ها است. در [43]، سیستم به عنوان شبکه ترکیبی دسته‌بندی شده است که در آن میانگین استفاده از محاسبه ارزش برای محاسبه طول صف‌ها و زمان پاسخ، بازده و استفاده از سیستم، مدل‌سازی شده است. روش بهینه‌سازی تکراری بر پایه درخت جستجوی باینری در

[1] . Kalman

تلاش است تا تعداد ماشین‌های مجازی مورد نیاز در هر سطر را بدون نقض آستانه عملکرد به حداقل برساند. در [57]، ابتدا مسأله بهینه‌سازی تعداد ماشین‌های مجازی برای به حداکثر رساندن سود پیش از ایجاد راه‌حل پویای قوانین ارتجاعی برنامه‌ریزی شد. در [27] ابتدا قاعده‌ای برای تعیین این موضوع مطرح شد که آیا اقدام مقیاس‌پذیری مورد نیاز است، و اگر این چنین باشد، چه نوع مسأله دسته‌بندی حل می‌شود. [36] نیز از قوانین برای تشخیص تغییرات حجم کار استفاده کرد و سپس الگوریتمی برای تصمیم‌گیری درباره حذف و اضافه ماشین‌های مجازی در هر لایه برنامه چندسطحی ارائه شد به طوری که زمان پاسخ برنامه کمتر از آستانه مشخص شده باشد و هزینه توسعه به حداقل برسد. یکی دیگر از ترکیب‌های مکانیسم در [38] ارائه شد، که در آن روش واکنشی بر پایه قواعدی برای ارزیابی منابع مورد استفاده قرار گرفت، در حالی که روش پیش‌بینی دقیق‌تر بر پایه مدل‌های رگرسیون (مدل سیستم) مقیاس داخلی مورد استفاده قرار گرفت. [61] بر پایه قوانین و مدل‌های پیش‌بینی شده است و ویژگی جالب آن پیش‌بینی خطا است که به طور مستقیم با روش لایه سازگار تطابق دارد.

آخرین دسته روش‌های ترکیبی آنهایی هستند که یادگیری ماشین را با تئوری بهینه‌سازی یا کنترل ترکیب می‌کنند. در [29]، نیازهای منابع به طور مستمر با توجه به حجم کار مورد نیاز تخمین زده شد. حجم کار با استفاده از روش تقریبی چندجمله‌ای پیش‌بینی شده و سپس به مجموعه‌ای از دسته‌های بار کاری طبقه‌بندی شد. سپس روش دومرحله‌ای اجرا شد. اول، اندازه ماشین مجازی بهینه (یعنی، مقدار CPU و RAM) و جریان متناظر مشخص می‌شود، بنابراین لازم است پتانسیل مقیاس عمودی مشخص شود. در مرحله دوم، تعداد ماشین‌های مجازی بهینه با اندازه مشخص شده محاسبه می‌شود، بنابراین لازم است پتانسیل مقیاس افقی مشخص شود. در [58]، شبکه‌های عصبی برای برآورد زمان تولید و پاسخ سیستم استفاده شدند و سپس کنترل کننده، مسأله بهینه‌سازی مقید را برای تعیین پیکربندی منابع بهینه از لحاظ تعداد ماشین مجازی و درجه تکرار داده‌ها حل کرد. در [15]، کنترل کننده پیش فرض مورد استفاده قرار گرفت که بر میزان کار نظارت دارد و از مدل رگرسیون برای این پیش‌بینی استفاده شد که آیا بار کاری باعث نقض توافق سطح خدمات می‌شود یا خیر، و بر این اساس واکنش نشان داده شد. این کنترل کننده با کنترل کننده بازخورد ترکیب می‌شود تا عملکرد را کنترل کند و بر پایه میزان انحراف از عملکرد دلخواه

تعیین شده در سطح خدمات هدف (SLO) واکنش نشان می‌دهد. در پایان، در [53] رفتار سیستم برای بار خارجی مفروض در گروه‌های نمایشی خوشه‌بندی شد. این کمک می‌کند تا مدل‌های توصیفی مقیاس افقی در قالب فرایندهای تصمیم‌گیری مارکوف مورد بررسی قرار گیرند که به طور بهینه حل می‌شوند. ویژگی منحصر به فرد این کار بررسی مدل موازی به کار رفته است تا تضمین‌های احتمالی را درباره عملکرد مورد انتظار اقدامات ارتجاعی ارائه دهد.

3.4. عمل ارتجاعی

اکثر مقالات درباره ارتجاع تنها مقیاس افقی را در نظر می‌گیرند، که در آنها تعداد مدل‌های ماشین‌های مجازی در عمل اصلاح می‌شود، به عنوان مثال [18، 22، 45، 53، 56، 64]. همچنین مقالاتی وجود دارند که تنها مقیاس عمومی را مورد استفاده قرار می‌دهند، مانند پیکربندی پویای CPU [40] و اندازه RAM و دیسک [62]. همچنین روش تکمیلی را برای حرکت زنده ماشین مجازی [50] ارائه دادیم و مقاله‌ای را درباره حرکت برنامه کاربردی [31] ارائه کردیم، که در آن تنها پایگاه‌های داده خاص به جای ماشین مجازی حرکت می‌کنند. با این وجود، روش‌هایی وجود دارد که کارهای مختلف را ترکیب می‌کنند. مقیاس افقی همراه با تنظیم مجدد برنامه در [23، 27 و 58] در نظر گرفته شد. بازسازی تنظیم شده میان پیشنهادات متفاوت است. به طور خاص، در [58] درجه پویایی تکرار می‌شود. در [23]، میزان حافظه پنهان به طور پویا کنترل می‌شود، در حالی که [27] حداکثر تعداد تفکیک داده‌ها را در هر گره در نظر گرفت. [29، 37 و 68] مقیاس افقی و عمودی را ترکیب کردند. [61 و 63] از روش‌های مورد استفاده برای تعداد ثابت ماشین مجازی برای ترکیب مقیاس عمودی و حرکت زنده استفاده کردند. مقیاس افقی نیز با حرکت زنده [54] و حرکت برنامه کاربردی [59] همراه است. در پایان، دو مقاله وجود دارد که سه نوع انعطاف‌پذیری منابع را ترکیب کردند. [20] مقیاس افقی و پایگاده داده‌های ارتباطی و حرکت زنده ماشین مجازی استفاده می‌کند، در حالی که [66] از مقیاس افقی، مقیاس عمودی (یعنی، پیکربندی CPU و RAM) و تنظیم مجدد برنامه (به عنوان مثال، تغییرات معماری برنامه کاربردی) استفاده کرد.

3.5. ارائه‌دهنده

اکثر مقالات درباره ارائه‌دهنده ابری تنها از ارائه‌دهنده عمومی یا خصوصی پشتیبانی می‌کنند. برخی از مقالات از بیش از یک ارائه‌دهنده پشتیبانی می‌کنند اما همزمان نیستند. به طور خاص، تمام این مقالات با Amazon-EC2 سازگار هستند و از Grid5000 (بنا بر [60])، پلت فرم مبتنی بر Nimbus (بنا بر [49])، OpenNebula (بنا بر [66])، Eucalyptus (بنا بر [38])، DAS-4 (یعنی، سیستم چندرسانه‌ای که توسط دانشگاه‌های هلند میزبانی شده است، بنا بر [32]) و OpenStack (بنا بر [41، 45 و 64])، پشتیبانی می‌کنند. در پایان، مقالاتی وجود دارند که شامل ارتجاع ارائه‌دهندگان چند سرویس‌دهنده همزمان هستند، مانند [55 و 56] که در بسیاری از زیرساخت‌های ابر خصوصی و عمومی ارائه می‌شوند.

3.6. ارزیابی

همان‌طور که در جدول 1 نشان داده شد، بیشتر مقالات از ارزیابی پیشنهادی خود استفاده می‌کنند، معیار RUBiS [11] در [35، 38، 40، 54 و 68] و TCP-[C/W] در [13، 36، 37 و 60] مورد استفاده قرار گرفت. یکی دیگر از معیارهای عمومی YCSB [24] است که در [15، 20، 23، 41، 45 و 64] مورد استفاده قرار گرفت. به علاوه در [60] از Apache Hadoop [1] با معیار مناسب MRBS [6] استفاده شد. در برخی از مقالات از TPC و YCSB [2 و 59] یا RUBiS و سیستم IBM [34، 61 و 63] استفاده کردند. CloudStone [3] در [29 و 47] مورد استفاده قرار گرفت؛ کاربرد بعدی Olio web 2.0 toolkit [9] CloudStone در ترکیب با نتیجه بار کاری Faban [4] است. MediaWiki [7] با معیار WikiBench [14] در [32] و ابزار میکرومعیار FIO [5] در [19] در ترکیب با اثر USR-1 اثرات MSR [8] مورد استفاده قرار گرفت. Apache Jmeter [2] در [39] مورد استفاده قرار گرفت. در پایان، مقالات دیگری وجود دارند که از معیارهای سفارشی استفاده می‌کنند مانند [21، 31، 43 و 58].

همچنین مقالاتی وجود دارند که با استفاده از شبیه‌سازی ارائه شده‌اند [16، 17، 22، 51 و 52]، به عنوان مثال، با استفاده از مجموعه آماری R [10]، یا OMNeT++ [67]. همچنین شبیه‌سازی‌ها می‌توانند معیارهایی را در نظر بگیرند، مانند [12]. در پایان، [53 و 57] از شبیه‌سازی استفاده کردند. [57] از تأثیرات جام جهانی فوتبال 1998 و سیاست پرداخت Amazon EC2 استفاده کرد، در حالی که [53] از تأثیرات حقیقی خوشه Cassandra NoSQL استفاده کرد.

3.7. بررسی و چالش‌های تحقیقاتی

در این بررسی، جنبه‌های تحلیل، برنامه‌ریزی و اجرای پیشرفته‌ترین ابر ارتجاعی را مورد بررسی قرار دادیم. در مرحله تحلیل، ارزیابی بازیابی مورد استفاده قرار گرفت تا وضعیت فعلی سیستم (به عنوان مثال، آیا از آن استفاده نشده یا بیش از حد استفاده شده است) و یا برآورد بار تغییرات بار آتی را بررسی کردیم. در مورد اول، روش عملی برای تعیین وضعیت کنونی سیستم از قوانین مبتنی بر آستانه استفاده می‌کند. با این وجود، مشخص کردن چنین قواعدی ساده نیست، زیرا به نیازهای برنامه و خواسته‌های سیستم مدیریت اجرایی بستگی دارد. برای غلبه بر این نگرانی، روش‌های مختلفی مانند ویژگی‌های قانون فازی ارائه شده است که دانش ذی‌نفعان آن را تحلیل و ذخیره می‌کند، و به کاربر اجازه می‌دهد تا آستانه‌های سطح بالایی را تعیین کنند که به طور خودکار بر پایه قواعد مبتنی بر آستانه مشخص می‌شوند. روش پیشنهادی دیگر تغییر توافق سطح خدمات و سطح خدمات هدف برای آستانه بر پایه قواعد است که از زبان‌های مشخص توافق سطح خدمات و سطح خدمات هدف سفارشی و سازگاری حل مسأله حاکم استفاده می‌کند. درباره پیش‌بینی بار آتی، روش‌های متعددی وجود دارند که با اشتباهات پیش‌بینی در ارتباطند. مقالاتی وجود دارند که سعی در محدود کردن خطای پیش‌بینی و یا در نظر گرفتن اشتباهات دارند. همچنین پیشنهاداتی وجود دارند که از بیش از یک الگوریتم استفاده می‌کنند تا مناسب‌ترین مکانیزم انتخابی را بر پایه بار کاری فعلی انتخاب کنند.

دانستن وضعیت فعلی سیستم و یا تغییر بار آتی باید در مرحله برنامه ریزی تصمیم‌گیری ارتجاعی حقیقی انجام شود. با این وجود، این گام مشکلاتی مانند تصمیم‌گیری میان مقیاس داخلی یا خارجی، انتخاب نوع ارتجاع مناسب یا تعیین میزان مقیای را نیز پنهان می‌کند. برای مقابله با این تصمیمات، روش‌های مختلفی از قبیل قوانین از پیش تعیین شده (یعنی ساده‌ترین شکل برنامه‌ریزی)، توابع مفید، مدل‌های سیستم، مکانیزم پیش‌بینی، روش‌های یادگیری ماشین و یا هر ترکیبی از قبل مورد استفاده قرار می‌گیرد تا رفتار سیستم را پیش از اجرای تصمیم واقعی بررسی کند. هر روش نقایص خاص خود را دارد، همان‌طور که در ادامه بررسی می‌شود:

- استفاده از قوانین از پیش تعیین شده، انعطاف‌پذیری نرم‌افزار را محدود می‌کند، زیرا مقدار و نوع مقیاس از قبل تعیین شده است. برای مقابله با این نگرانی، ویژگی‌های قوانین پویا و یا به روزرسانی قواعد در مقالات ارائه شده است.

- بهینه‌سازی تابع سود که شامل معیارهای انتخاب شده و وزن مناسب است و می‌تواند منجر به تناقض میان گزینه‌ها شود، با این وجود مشخصات این تابع مستلزم دانش اجرایی ویژه است. برای غلبه بر این نگرانی، توابع ثابتی ارائه شده‌اند که به اندازه کافی کلی هستند و برای بسیاری از سیستم‌ها کاربرد دارند.

- ارائه مشخصات مدل دستگاه با وظیفه غیربدیهی برای ایجاد مدل قابل اعتمادی که متغیرهای ورودی و خروجی سیستم را طراحی می‌کند، مشکل است. علاوه بر این، مدل سیستم مانع از انعطاف‌پذیری مکانیزم کششی می‌شود، بنابراین پس از تغییر ساختار سیستم، مدل به بازسازی نیاز دارد.

- روش‌هایی که از پیش‌بینی رفتار سیستم استفاده می‌کنند از اشتباهات پیش‌بینی شده رنج می‌برند. راه‌حل پیشنهادی مشابه راه‌حل‌هایی است که برای مرحله تحلیل مورد استفاده قرار می‌گیرند.

- روش امیدوارکننده استفاده از یادگیری ماشین است که در آن مکانیسم ارتجاعی پیش از توسعه واقعی آموزش می‌بینند. مرحله آموزش نیز می‌تواند در هنگام توسعه حقیقی اعمال شود که برای آموزش پویا مجاز است. با این وجود، در فاز بعدی، ممکن است مکانیسم قادر به اداره ارتجاع به خوبی ابتدای توسعه نباشد. برای پرهیز از تصمیمات اشتباه قید پایین یا بالای سیستم، روش‌ها به استفاده از مکانیزم واکنشی آستانه‌ای تا زمانی تمایل دارند که مکانیسم به خوبی آموزش ببیند و قادر به مدیریت ارتجاعی کارای باشد. بحث جالب در این رابطه در [28] نیز ارائه شده است.

در مرحله اجرایی، تصمیم‌گیری ارتجاعی حقیقی از طریق مدیریت ارزیابی مدیریت ارتجاعی انجام می‌شود. مدیریت ارتجاعی، مکانیسم استاندارد را توسط ارائه‌دهندگان ابر به عنوان خدمات یا از طریق API راه دور (به عنوان مثال، سرویس مقیاس خودکار Amazon EC2) یا مدیر خارجی ارائه می‌دهد.

معیارهای مورد نیاز تحقیقات بیشتر. به عنوان نکته نهایی، اگرچه مجموعه‌ای از روش‌های انعطاف‌پذیر زیادی وجود دارد و مقدار قابل توجهی از آنها با اهداف متعددیمواجه می‌شوند، هیچ راه‌حل سیستماتیکی برای بهینه‌سازی چندهدفه چندگانه تحت چند هدف متضاد، به عنوان مثال، تضمین بهینگی پارتو، ارائه نشده است. معتقدیم که این می‌تواند موضوع جالب کارهای بعدی باشد. موضوع جالب دیگر ارائه چارچوب‌هایی است که می‌توانند چندین راه‌حل را که اکنون از هم جدا شده‌اند، با هم ترکیب کنند. در پایان، برای ارزیابی بهتر کیفیت هر یک از روش‌ها مستلزم تحقیقات بیشتر درباره معیارها است.

4. خلاصه

این بررسی با هدف طبقه‌بندی و ارائه خلاصه‌ای از چندین روش برای کشف ابر ارتجاعی امروزی انجام شده است. طبقه‌بندی را در طیف گسترده‌ای از جنبه‌های پوشش داده شده ارائه دادیم و جزئیات بیشتر را برای هر یک از جنبه‌ها و چالش‌های عمده تحقیق بررسی کردیم. در پایان، موضوعاتی را پیشنهاد دادیم که به تحقیق بیشتری نیاز دارند.

References

1. Apache hadoop. https://hadoop.apache.org/. Accessed 11 Jun 2015
2. Apache jmeter: Graphical server performance testing tool. http://jmeter.apache. org/. Accessed 11 Jun 2015
3. Cloudstone. http://parsa.epfl.ch/cloudsuite/web.html. Accessed 11 Jun 2015
4. Faban: Performance workload creation and execution framework. http://faban. org/. Accessed 11 Jun 2015
5. Fio: A micro-benchmarking tool. http://freshmeat.net/projects/fio. Accessed 11 Jun 2015
6. Hadoop mapreduce dependability, performance benchmarking. http://sardes. inrialpes.fr/research/mrbs/. Accessed 11 Jun 2015
7. Mediawiki: Web hosting benchmark. http://www.wikibench.eu. Accessed 11 Jun 2015
8. Msr cambridge traces. http://iotta.snia.org/traces/388. Accessed 11 Jun 2015
9. Olio web 2.0 toolkit. http://incubator.apache.org/projects/olio.html. Accessed 11 Jun 2015
10. The r project for statistical computing. http://www.r-project.org. Accessed 11 Jun 2015
11. Rubis: Rice university bidding system. http://rubis.ow2.org. Accessed 11 Jun 2015
12. Specjenterprise benchmark system. https://www.spec.org/jEnterprise2010/. Accessed 11 Jun 2015
13. Tcp. http://www.tpc.org. Accessed 11 Jun 2015
14. Wikibench: Web hosting benchmark. http://www.wikibench.eu. Accessed 11 Jun 2015
15. Al-Shishtawy, A., Vlassov, V.: Elastman: elasticity manager for elastic key-value stores in the cloud. In: ACM Cloud and Autonomic Computing Conference, CAC 2013, Miami, FL, USA, 5–9 August 2013, p. 7 (2013)
16. Ali-Eldin, A., Tordsson, J., Elmroth, E.: An adaptive hybrid elasticity controller for cloud infrastructures. In: 2012 IEEE Network Operations and Management Symposium (NOMS), pp. 204–212 (2012)
17. Almeida Morais, F., Vilar Brasileiro, F., Vigolvino Lopes, R., Araujo Santos, R., Satterfield, W., Rosa, L.: Autoflex: service agnostic auto-scaling framework for IaaS deployment models. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 42–49 (2013)
18. Ashraf, A., Byholm, B., Porres, I.: Cramp: cost-efficient resource allocation for multiple web applications with proactive scaling. In: 2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom), pp. 581– 586 (2012)
19. Bairavasundaram, L.N., Soundararajan, G., Mathur, V., Voruganti, K., Srinivasan, K.: Responding rapidly to service level violations using virtual appliances. SIGOPS Oper. Syst. Rev. 46(3), 32–40 (2012)
20. Barker, S.K., Chi, Y., Hacig¨um¨us, H., Shenoy, P.J., Cecchet, E.: Shuttledb: database-aware elasticity in the cloud. In: 11th International Conference on Autonomic Computing, ICAC 2014, Philadelphia, PA, USA, 18–20 June 2014, pp. 33–43 (2014)
21. Beernaert, L., Matos, M., Vila¸ca, R., Oliveira, R.: Automatic elasticity in openstack. In: Proceedings of the Workshop on Secure and Dependable Middleware for Cloud Monitoring and Management, p. 2 (2012)
22. C´amara, J., Moreno, G.A., Garlan, D.: Stochastic game analysis and latency awareness for proactive self-adaptation. In: SEAMS, pp. 155–164 (2014)
23. Chalkiadaki, M., Magoutis, K.: Managing service performance in the cassandra distributed storage system. In: IEEE 5th International Conference on Cloud Computing Technology and Science, CloudCom 2013, Bristol, UK, 2–5 December 2013, vol. 1, pp. 64–71 (2013)
24. Cooper, B.F., Silberstein, A., Tam, E., Ramakrishnan, R., Sears, R.: Benchmarking cloud serving systems with YCSB. In: Proceedings of the 1st ACM Symposium on Cloud Computing, pp. 143–154 (2010)
25. Copil, G., Moldovan, D., Truong, H.L., Dustdar, S.: On controlling cloud services elasticity in heterogeneous clouds. In: 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC), pp. 573–578 (2014)
26. Coutinho, E.F., de Carvalho Sousa, F.R., Rego, P.A.L., Gomes, D.G., de Souza, J.N.: Elasticity in cloud computing: a survey. Ann. Telecommun.-Annales des T´el´eCommuni 70, 289–309 (2015)

27. Cruz, F., Maia, F., Matos, M., Oliveira, R., Paulo, J., Pereira, J., Vila¸ca, R.: Met: workload aware elasticity for NoSQL. In: Eighth Eurosys Conference 2013, EuroSys 2013, Prague, Czech Republic, 14–17 April 2013, pp. 183–196 (2013)
28. Dutreilh, X., Rivierre, N., Moreau, A., Malenfant, J., Truck, I.: From data center resource allocation to control theory and back. In: IEEE CLOUD, pp. 410–417 (2010)
29. Dutta, S., Gera, S., Verma, A., Viswanathan, B.: Smartscale: automatic application scaling in enterprise clouds. In: IEEE CLOUD. pp. 221–228 (2012)
30. Elmore, A.J., Das, S., Agrawal, D., El Abbadi, A.: Zephyr: live migration in shared nothing databases for elastic cloud platforms. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, pp. 301–312 (2011)
31. Elmore, A.J., Das, S., Pucher, A., Agrawal, D., El Abbadi, A., Yan, X.: Characterizing tenant behavior for placement and crisis mitigation in multitenant DBMSS, pp. 517–528 (2013)
32. Fernandez, H., Pierre, G., Kielmann, T.: Autoscaling web applications in heterogeneous cloud infrastructures. In: 2014 IEEE International Conference on Cloud Engineering, pp. 195–204 (2014)
33. Galante, G., de Bona, L.C.E.: A survey on cloud computing elasticity. In: 2012 IEEE Fifth International Conference on Utility and Cloud Computing (UCC), pp. 263–270 (2012)
34. Gedik, B., Andrade, H., Wu, K.L., Yu, P.S., Doo, M.: SPADE: the system s declarative stream processing engine. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 1123–1134 (2008)
35. Gueye, S.M.K., Palma, N.D., Rutten, E., Tchana, A., Berthier, N.: Coordinating ´ self-sizing and self-repair managers for multi-tier systems. Future Gener. Comp. Syst. 35, 14–26 (2014)
36. Han, R., Ghanem, M., Guo, L., Guo, Y., Osmond, M.: Enabling cost-aware and adaptive elasticity of multi-tier cloud applications. Future Gener. Comp. Syst. 32, 82–98 (2014)
37. Han, R., Guo, L., Ghanem, M.M., Guo, Y.: Lightweight resource scaling for cloud applications. In: 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 644–651 (2012)
38. Iqbal, W., Dailey, M.N., Carrera, D., Janecek, P.: Adaptive resource provisioning for read intensive multi-tier applications in the cloud. Future Gener. Comput. Syst. 27(6), 871–879 (2011)
39. Jamshidi, P., Ahmad, A., Pahl, C.: Autonomic resource provisioning for cloudbased software. In: Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-managing Systems, SEAMS 2014, Hyderabad, India, 2–3 June 2014, pp. 95–104 (2014)
40. Kalyvianaki, E., Charalambous, T., Hand, S.: Self-adaptive and self-configured CPU resource provisioning for virtualized servers using Kalman filters. In: Proceedings of the 6th International Conference on Autonomic Computing, ICAC 2009, 15–19 June 2009, Barcelona, Spain, pp. 117–126 (2009)
41. Kassela, E., Boumpouka, C., Konstantinou, I., Koziris, N.: Automated workloadaware elasticity of NoSQL clusters in the cloud. In: 2014 IEEE International Conference on Big Data, Big Data 2014, Washington, DC, USA, 27–30 October 2014, pp. 195–200 (2014)
42. Katukoori, V.K.: Standardizing Availability Definition. University of New Orleans, New orleans (1995)
43. Kaur, P.D., Chana, I.: A resource elasticity framework for QOS-aware execution of cloud applications. Future Gener. Comp. Syst. 37, 14–25 (2014)
44. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. IEEE Comput. 36(1), 41–50 (2003)
45. Konstantinou, I., Angelou, E., Tsoumakos, D., Boumpouka, C., Koziris, N., Sioutas, S.: Tiramola: elastic NoSQL provisioning through a cloud management platform. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 725–728. ACM (2012)
46. Li, Z., Zhang, H., Obrien, L., Cai, R., Flint, S.: On evaluating commercial cloud services: a systematic review. J. Syst. Softw. 86, 2371–2393 (2013)
47. Lim, H.C., Babu, S., Chase, J.S.: Automated control for elastic storage. In: Proceedings of the 7th International Conference on Autonomic Computing, pp. 1–10 (2010)
48. Lorido-Botran, T., Miguel-Alonso, J., Lozano, J.A.: A review of auto-scaling techniques for elastic applications in cloud environments. J. Grid Comput. 12(4), 559– 592 (2014)

49. Marshall, P., Keahey, K., Freeman, T.: Elastic site: using clouds to elastically extend site resources. In: CCGRID, pp. 43–52 (2010)
50. Mastroianni, C., Meo, M., Papuzzo, G.: Probabilistic consolidation of virtual machines in self-organizing cloud data centers. IEEE Trans. Cloud Comput. 1(2), 215–228 (2013)
51. Moore, L., Bean, K., Ellahi, T.: A coordinated reactive and predictive approach to cloud elasticity. In: The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization, CLOUD COMPUTING 2013, pp. 87–92 (2013)
52. Moore, L.R., Bean, K., Ellahi, T.: Transforming reactive auto-scaling into proactive auto-scaling. In: Proceedings of the 3rd International Workshop on Cloud Data and Platforms, pp. 7–12 (2013)
53. Naskos, A., Stachtiari, E., Gounaris, A., Katsaros, P., Tsoumakos, D., Konstantinou, I., Sioutas, S.: Dependable horizontal scaling based on probabilistic model checking. In: CCGrid (2015)
54. Nguyen, H., Shen, Z., Gu, X., Subbiah, S., Wilkes, J.: AGILE: elastic distributed resource scaling for infrastructure-as-a-service. In: 10th International Conference on Autonomic Computing, ICAC 2013, San Jose, CA, USA, 26–28 June 2013, pp. 69–82 (2013)
55. Paraiso, F., Merle, P., Seinturier, L.: Managing elasticity across multiple cloud providers. In: Proceedings of the 2013 International Workshop on Multi-cloud Applications and Federated Clouds, pp. 53–60 (2013)
56. Paraiso, F., Merle, P., Seinturier, L.: soCloud: a service-oriented component-based PaaS for managing portability, provisioning, elasticity, and high availability across multiple clouds. CoRR abs/1407.1963 (2014)
57. Perez-Palacin, D., Mirandola, R., Calinescu, R.: Synthesis of adaptation plans for cloud infrastructure with hybrid cost models. In: 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications, Verona, Italy, 27–29 August 2014, pp. 443–450 (2014)
58. di Sanzo, P., Rughetti, D., Ciciani, B., Quaglia, F.: Auto-tuning of cloud-based in-memory transactional data grids via machine learning. In: Second Symposium on Network Cloud Computing and Applications, NCCA 2012, London, UK, 3–4 December 2012, pp. 9–16 (2012)
59. Serafini, M., Mansour, E., Aboulnaga, A., Salem, K., Rafiq, T., Minhas, U.F.: Accordion: elastic scalability for database systems supporting distributed transactions. PVLDB 7(12), 1035–1046 (2014)
60. Serrano, D., Bouchenak, S., Kouki, Y., Ledoux, T., Lejeune, J., Sopena, J., Arantes, L., Sens, P.: Towards QOS-oriented sla guarantees for online cloud services. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 50–57 (2013)
61. 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, pp. 5:1–5:14 (2011)
62. da Silva Dias, A., Nakamura, L.H.V., Estrella, J.C., Santana, R.H.C., Santana, M.J.: Providing IaaS resources automatically through prediction and monitoring approaches. In: IEEE Symposium on Computers and Communications, ISCC 2014, Funchal, Madeira, Portugal, 23–26 June 2014, pp. 1–7 (2014)
63. Tan, Y., Nguyen, H., Shen, Z., Gu, X., Venkatramani, C., Rajan, D.: Prepare: predictive performance anomaly prevention for virtualized cloud systems. In: 2012 IEEE 32nd International Conference on Distributed Computing Systems (ICDCS), pp. 285–294 (2012)
64. Tsoumakos, D., Konstantinou, I., Boumpouka, C., Sioutas, S., Koziris, N.: Automated, elastic resource provisioning for NoSQL clusters using tiramola. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 34–41 (2013)
65. Uhlig, R., Neiger, G., Rodgers, D., Santoni, A.L., Martins, F., Anderson, A.V., Bennett, S.M., K¨agi, A., Leung, F.H., Smith, L.: Intel virtualization technology. Computer 38(5), 48–56 (2005)
66. Vaquero, L.M., Mor´an, D., Gal´an, F., Alcaraz-Calero, J.M.: Towards runtime reconfiguration of application control policies in the cloud. J. Netw. Syst. Manage. 20(4), 489–512 (2012)
67. Varga, A., Hornig, R.: An overview of the OMNeT++ simulation environment. In: Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, p. 60. ICST (2008)
68. Vasic, N., Novakovic, D.M., Miucin, S., Kostic, D., Bianchini, R.: Dejavu: accelerating resource allocation in virtualized environments. In: ASPLOS, pp. 423–436 (2012)