Archive of SID

1st International Conference on new achievements in Science & Technology
24th August 2017
اولین کنفرانس بین المللی دستاوردهای نوین در علوم و تکنولوژی
۲ شهریور ماه ۱۳۹۶ – تهران

به کارگیری الگوریتم ترکیبی بهینه سازی رقابت استعماری و الگوریتم ماهی‌های مصنوعی جهت کشف بازه های زمانی مناسب در SLA سرویس‌های رایانش ابری

بهناز معینی[۱]، علی اصغر صفایی[۲]

۱ دانشجوی کارشناسی ارشد، گروه کامپیوتر- واحد بین الملل قشم- دانشگاه آزاد اسلامی، قشم- ایران

moeini_behnaz@yahoo.com

۳ استادیار، گروه انفورماتیک پزشکی- دانشگاه تربیت مدرس، تهران- ایران

aa.safaei@modares.ac.ir

چکیده

محاسبات ابری در اجرای زیرساخت‌های محاسباتی مقیاس‌پذیر براساس تقاضا به یک محیط شناخته‌شده تبدیل‌شده‌اند. زیرساخت‌های ابر خود مدیریتی، به منظور تطابق با نیازهای کاربران که توسط توافق‌نامه سطح سرویس تعریف شده‌اند و به حداقل رساندن تعامل کاربر با محیط محاسباتی مورد نیاز هستند. بنابراین راهبردهای نظارتی کافی و تشخیص به موقع از نقض توافق‌های محتمل مسائل پژوهشی چالش‌برانگیز می‌باشند.

منابع و خدمات ابری، براساس توافقنامه سطح خدمات ارائه می‌شوند. این توافق شرایط استفاده و جرائم را درصورت وجود تخطی، بیان می‌کند. باوجود اینکه عملیات بزرگی در حوزه نظارت و قانون‌مندی توافقنامه سطح خدمات در زیرساخت و لایه‌های بستر صورت گرفته است، اما توافقنامه‌های سطح خدمات عموماً در لایه کاربرد، تضمین می‌شوند. بدین منظور یک معماری نظارتی اپلیکیشن به نامCASViD ارائه‌شده است. معماریCASViD تخطی‌ها را در لایه اپلیکیشن و در ابزار ابر، بررسی و کشف می‌کند. روش دیگری برای کشف خطا، معماری زیرساخت تشخیص نقض توافق سطح سرویس بنام DeSVi می‌باشد که نقض‌های توافق‌نامه را از طریق نظارت بر منابع پیچیده تشخیص می‌دهد. براساس درخواست کاربر، DeSVi منابع محاسباتی را برای یک سرویس درخواست شده اختصاص و ترتیب استقرار خود را در یک محیط مجازی می‌دهد. همچنین با استفاده از چارچوب LoM2HiS قادر به نگاشت بین معیارهای سطح پایین منابع به معیارهای سطح توافقنامه می‌شوند. تشخیص نقض توافقنامه متکی بر اهداف سطح خدمات از پیش تعریف‌شده و استفاده از پایگاه داده‌های دانش جهت مدیریت و جلوگیری از چنین تخلفاتی است. این معماری‌ها دوره نظارت را براساس بازه‌های زمانی ایستا مشخص می‌کنند که در تحقیق پیش رو، قصد بر این است تا با استفاده از الگوریتم ترکیبی رقابت استعماری و **الگوریتم ماهی‌های مصنوعی** دوره‌های نظارت را به صوت پویا کنترل نماییم.

واژه‌های کلیدی: محاسبات ابری، توافق‌نامه سطح خدمات، معماری CASViD، معماری DeSVi ، چارچوب LoM2HiS

www.SID.ir

Archive of SID

1st International Conference on new achievements in Science & Technology
24th August 2017
اولین کنفرانس بین المللی دستاوردهای نوین در علوم و تکنولوژی
۲ شهریور ماه ۱۳۹۶ – تهران

Implementing a hybrid optimization algorithm for ICA and AFSA to discover the appropriate time intervals in SLA cloud computing services.

behnaz,moeini [1]; Ali Asghar,safaei[2]

1- Department of Computer, Qeshm international Branch, Islamic Azad University, qeshm ,Iran

Email: *moeini_behnaz@yahoo.com*

2- Department of Medical Informatics, Tarbiat Modares University, Tehran-Iran

Email: *aa.safaei@modares.ac.ir*

Abstract

In implementing scalable cloud computing infrastructure on demand has become a known environment. Their cloud infrastructure management to comply with the needs of users that have been defined by service level agreements and to minimize user interaction with the computing environment required Therefore, adequate regulatory strategies and detection of possible violations of the agreement are challenging research problems.

Sources and cloud services based on service level agreements are provided. If there is a breach of this agreement, terms of use and crime, states Although large-scale operations in the area of infrastructure and service level agreement monitoring and lawful substrate layer has been done, but generally the service level agreements at the application layer, Guarantee. An app called CASViD regulatory architecture is presented. CASViD architecture violations in applications layer and cloud tools to investigate and discovers. Another method for error detection, diagnosis infrastructure architecture, called service level agreement violations liable DeSVi The breach of the agreement by monitoring detects complex. Based on user requests, DeSVi computing resources assigned to a requested service and the way in a virtual environment offers. Using LoM2HiS framework capable of mapping between low-level metrics to measure the level of agreement on a pick. Agreement defect detection based on pre-defined service level objectives and using the knowledge database is used to manage and prevent such violations. This architecture is based on the closed interval specified static moved us to compete with the algorithm combines ICA and AFSA to dynamically control will sound.

Keywords: *Cloud computing, service level agreements, architecture CASViD, Artificial Fish Swarm Algorithm, architecture DeSVi, the LoM2HiS*

www.SID.ir

Archive of SID

۱- مقدمه

در این پژوهش سعی شده است تا به بررسی الگوریتم‌های فرامکاشفه‌ای و تخطی در محاسبات ابری پرداخته شود. انجام محاسبات ابری در یک معماری DeSVi از توافقنامه سطح سرویس با شبیه سازی این معماری میسر شده است. محاسبات ابری کاربردهای زیادی دارد که ملزومات کلی طرف کاربر پیچیدگی آن را کاهش می‌دهد (زسیس و همکاران[1]،۲۰۱۲). مدیریت قابل‌اطمینان انعطاف‌پذیر توافق‌های سطح سرویس، برای مشتری و ارائه‌دهنده هر دو، حائز اهمیت می‌باشند. از یک سو، باعث جلوگیری از تخطی‌های توافقنامه سطح سرویس و به تبع آن موجب جلوگیری از مجازات برای ارائه‌دهنده سرویس خواهد شد و از سوی دیگر براساس واکنش به موقع و قابل‌انعطاف به تخطی‌های توافقنامه سطح سرویس، ارتباط کاربر با سیستم به حداقل کاهش می‌یابد، این امر باعث می‌شود رایانش ابری به‌عنوان یک قالب انعطاف‌پذیر و قابل‌اعتماد از رایانش موردتقاضا، محسوب گردد. الگوریتم‌های تکاملی[2]زیرمجموعه‌ای از محاسبات تکاملی است و در شاخه هوش مصنوعی قرار می‌گیرد.

۲- تعریف مسئله و بیان سؤال‌های اصلی تحقیق

پردازش ابری[3] امروزه به‌عنوان یک نمونه محاسباتی جدید که قصد تهیه یک محیط محاسباتی پویای قابل‌اطمینان، سفارشی و باکیفیت سرویس[4] ضمانت شده را دارد، به وجود آمده است. این نوع از محاسبات، سرویس‌های فناوری اطلاعات فراگیری را به کاربران سراسر جهان ارائه می‌کنند و براساس مدل پرداخت به ازای استفاده، قادر به میزبانی کاربردهای فراگیر در حوزه‌های تجاری، علمی و اقتصادی می‌باشند. این مدل محاسباتی باوجود ارائه سرویس‌های اشتراکی مزایای بسیاری در کاهش هزینه‌های عملیاتی همچون رفع نیاز به خرید نرم‌افزار یا سخت‌افزار توسط کاربران را فراهم نموده است. (کامار، سازنا[5]،۲۰۱۱).

پردازش ابری در کنار مزایای فراوانی که در اختیار کاربران قرار می‌دهد مشکلات و خطراتی چون امنیت، ارتباط ناامن، اشتراک منابع و حملات داخلی را نیز به همراه دارد. درصورت بروز خسارت و از دست رفتن داده‌ها وجود ضمانت‌نامه سطح کیفی خدمات[6] می‌تواند به کاربران ابر برای اثبات ادعای خود علیه ارائه‌دهنده سرویس ابر کمک کند (امیکارها، و همکاران،۲۰۱۲).

منظور از ضمانت‌نامه سطح کیفی خدمات یا Service Level Agreement (SLA) است که قرارداد تضمین و تأمین میزبانی معین از کیفیت سرویس از سوی سرویس‌دهنده برای سرویس‌گیرنده یا کاربر می‌باشد. به بیان دیگر مانند قرارداد رسمی بین فراهم آورنده خدمات[7] و مشتری[8] است و در برگیرنده اطلاعاتی از جمله مشخصات خدمات، نام و مقادیر است. این توافقنامه به عنوان یک سطح مورد انتظار در ارائه سرویس بین مصرف‌کننده و ارائه‌دهنده می‌باشد . نظارت کافی و کشف تخطی از این توافقنامه امری مهم و چالش‌برانگیز می‌باشد(امیکارها و همکاران،۲۰۱۲).

هدف ما در این تحقیق یافتن بهترین بازه زمانی به‌صورت پویا، جهت کنترل ضمانت‌نامه سطح کیفی خدمات و بهینه‌سازی تضاد منافع ناشی از هزینه تخطی‌های کشف نشده و هزینه ناشی از سربار سیستم و همچنین اختصاص بازه‌های زمانی پویا وابسته به میزان استفاده از منابع است، تا موجب بهینه شدن هزینه‌های نظارت بر عملکرد منابع و نگاشت معیارها شود. به دلایل زیر ترکیب دو الگوریتم رقابت استعماری و **الگوریتم ماهی‌های مصنوعی(AFSA)**[9] را به عنوان الگوریتم تکاملی در این تحقیق انتخاب نموده ام:**الگوریتم رقابت استعماری:** از ویژگی‌های این الگوریتم نو بودن ایده، مبتنی بر رفتار اجتماعی انسان که

[1] Zessis et al
[2] Evolutionary algorithms
[3] cloud computing
[4] Quality of Service
[5] S. Kumar, A. Saxena
[6] SLA
[7] Service Provider
[8] Customer
[9] AFSA

www.SID.ir

هوشمندانه‌تر از رفتارهای بیولوژیکی است، سرعت همگرایی بالا، توانایی بهینه‌سازی توابعی با تعداد متغیرهای بسیار زیاد، سرعت یافتن جواب بهینه، توانایی بهینه‌سازی خوب را می‌توان بیان کرد (آتش پز گرگری و همکاران ،۲۰۰۷).

الگوریتم دسته ماهی‌های مصنوعی: (AFSA) یک مدل محاسباتی تکاملی است که مبتنی بر الگوریتم‌های هوش جمعی و برگرفته‌شده از طبیعت می‌باشد. عملکرد این الگوریتم براساس جستجوی تصادفی است. این الگوریتم دارای خصوصیاتی ازجمله سرعت همگرایی بالا، حساس نبودن به مقادیر اولیه‌ی ماهی‌های مصنوعی، انعطاف‌پذیری و تحمل‌پذیری خطا می‌باشد که آن را برای حل مسائل بهینه‌سازی قابل‌قبول می‌کند(دکتر لیو و همکاران[1]،۲۰۰۲). لذا باتوجه به ویژگی‌های خوبی که الگوریتم رقابت استعماری و الگوریتم دسته ماهی‌ها دارند، ویژگی‌های این دو الگوریتم را ترکیب نموده و انتظار داریم در کشف تخطی‌های سطح سرویس پاسخی مناسب و قابل قبول ارایه نمایند.

۳- اهمیت و ضرورت انجام تحقیق

توافقنامه سطح سرویس به عنوان یک سطح مورد انتظار در ارائه سرویس بین مصرف‌کننده و ارائه‌دهنده می‌باشد. نظارت کافی و کشف تخطی از این توافقنامه امری مهم و چالش‌برانگیز بوده و به همین جهت در سال ۲۰۱۲ معماری ارائه گردید که کار نظارت و کشف تخطی از توافقنامه سطح سرویس را به شکل خودکار انجام می‌داد، این معماری DeSVi نامیده می‌شود (امیکارها و همکاران،۲۰۱۲). یکی از استراتژی‌های نظارتی برای رفع چالش‌های تخطی از توافقنامه این امر است که معیارهایی مانند سرعت پردازنده مرکزی، زمان آغاز و زمان خاتمه کار به مفاد توافقنامه سطح سرویس تبدیل شوند (نگاشت معیارها) تا ضمن ارائه سرویس نحوه تخصیص منابع هم کنترل شود. سعی کنیم که از رویکرد پویا برای مشخص کردن بازه‌های زمانی استفاده شود می‌توانیم بهترین بازه‌ها را انتخاب و این سنجش را بدون نیاز به دخالت مستقیم مدیران فضای ابری انجام دهیم.

۵-مرور ادبیات و سوابق مربوط به تحقیق

کارهایی که در زمینه کشف بازه‌های زمانی انجام شده محدود بوده و در زمینه‌های نظارت بر منابع، مدیریت توافقنامه سطح سرویس شامل؛ کشف تخطی و تکنیک‌های نگاشت معیارهای نظارت انجام‌شده است (امیکارها و همکاران،۲۰۱۲).

در مقاله مورد مطالعه دیگر، رویکردی جهت نگاشت توافقنامه سطح سرویس خصوصی به توافقنامه سطح سرویس استاندارد عمومی ارائه شده است. این رویکرد به کاربران این امکان را می‌دهد تا در سازمان‌های خصوصی الگوی توافقنامه خود را داشته باشند که برای پردازش‌های داخلی کاربرد دارند و در آن به محاسبات توری اشاره شده است (مارو امکارها، براندیک، التمن[2]،۲۰۱۲)

در مقاله مورد مطالعه دیگر، منابع محاسباتی براساس یک الگوی توافقنامه سطح سرویس تعریف می‌شوند، تخصیص منابع محاسبات ابری نه تنها به ملزومات علمی بلکه ملزومات غیرعلمی نیز وابسته است؛ که در این راستا امیکارها در سال ۲۰۱۲ معماری برای کشف خطاها ارائه داده است. آن‌ها در این کار معیارهای سطح پایین را به معیارهای سطح بالا نگاشت می‌دهند تا نحوه تخصیص منابع کنترل گردد. ولی این کار باعث ایجاد وقفه می‌شود و سرباری به سیستم وارد می‌کند و از طرف دیگر اگر زمان بین این نگاشت‌ها افزایش یابد ممکن است خطاها دیده نشوند. کار امیکارها مکانیزمی برای اختصاص پویای این بازه‌های نظارت ارائه نمی‌دهد. راک و همکاران، نظارت ابری یک برنامه را با استفاده از روش mOSAIC[3] ارائه کردند. در مرحله اول؛ نویسنده، توسعه برنامه مشخص‌شده را با استفاده از mOSAIC API[4] توصیف می‌کند که روی محیط ابری گسترش‌یافته است. در مرحله دوم، برخی فن‌های نظارت بر برنامه بیان‌شده است. هدف آن‌ها، تنها جمع‌آوری اطلاعاتی است که می‌تواند در اجرای دستی یا خودکار تنظیم بار[5] ، کاهش/ افزایش تعداد ماشین مجازی یا محاسبه هزینه کل اجرای برنامه‌ها، مورد استفاده قرار گیرد. روش آن‌ها، کشف تخطی از توافقنامه سطح

[1] Li Xiao Lei
[2] Maurer M, Emeakaroha V.C, Brandic I, Altmann J
[3] mOSAIC Is a Open Source API and Platform Multiple Cloud
[4] Aplication Programing Interface
[5] Load Balancing

www.SID.ir

سرویس به منظور اجتناب از هزینه در نظر گرفته نمی‌شود و به علاوه عمومی نیست. چون تنها برنامه‌هایی را کنترل می‌کند که با استفاده از mOSAIC API، توسعه‌یافته‌اند (راک و همکاران،۲۰۱۳).

ساروانان و همکاران، بر روی کاهش نقض تعهدات توافقنامه سطح سرویس به‌عنوان یک نیاز جهت بالا بردن کیفیت سرویس و راضی کردن مشتریان تمرکز کرده‌اند. روش Bee-MMT استفاده شده در آن مقاله باتوجه به کاهش مصرف برق باعث نقض بیشتر توافقنامه سطح سرویس نسبت به روش‌های دیگر شده است. لذا با استفاده از مدل کاهشی ADA[2] SALMON[1] به کنترل خودکار و تجزیه‌وتحلیل توافقنامه سطح سرویس از محدودیت‌های ارضای مسئله در آن مقاله پرداخته‌شده است (ساروانان و همکاران،۲۰۱۵).

سارا زنگنه و احمد فراهی، یک مکانیزم هوشمند جهت یافتن بازه‌های زمانی پویا جهت نظارت و بهینه کردن تضاد منافع در معماری Desvi ارائه دادند. فرآیند اندازه‌گیری شامل؛ نظارت بر همه ماشین‌های مجازی، پردازش داده‌های نظارت‌شده، نگاشت معیارهای سطح پایین به سطح بالا، توافقنامه سطح سرویس و ارزیابی تابع هدف است. رویکرد مقاله باتوجه به مصرف منابع و معیارهای سطح توافقنامه در هر بار تکرار اجرا، تعداد اندازه‌گیری متفاوت اما نزدیک را نشان می‌دهد. هرچه بازه‌های نظارتی نزدیک‌تر باشند احتمال اینکه نقضی بروز کند و دیده نشود کمتر خواهد بود. این مسئله به‌خوبی از نتایج قابل استنباط است. هرچه بازه‌های نظارتی کوتاه‌تر باشد تعداد نقض‌های توافقنامه‌ای که دیده نمی‌شوند کاهش می‌یابد، ولی سربار وارد شده به سیستم افزایش می‌یابد. نتایج نشان می‌دهد که رویکرد پیشنهادی توانسته به یک نتیجه بهینه بین بازه‌های کوتاه و بلند در معماری Desvi برسد ولی اطلاعات دقیقی در مورد پیاده‌سازی ارائه داده نشده است (سارا زنگنه و احمد فراهی،۱۳۹۳).

ناهید غفوری و فاطمه سعادت جو، با استفاده از الگوریتم رقابت استعماری تضاد منافع میان سربار سیستم و تخطی در محاسبات ابری جهت یافتن بازه‌های پویا در معماری Desvi را بهینه کرده‌اند. ازآنجا که در کار انجام شده توسط امیکارها و همکاران،۲۰۱۲ بازه‌های نظارتی به شکل استاتیک انجام‌گرفته است و این امر باعث می‌شد هنگامی‌که بازه‌ها کوتاه بودند سیستم نظارتی خوب عمل کرده و زمانی که بازه‌ها بلند می‌شدند تخطی‌ها دیده نشوند. برای جلوگیری از چنین رفتاری، رویکردی ارائه شد که به‌وسیله آن بازه‌ها باتوجه به مصرف منابع و معیارهای توافقنامه سطح سرویس به‌صورت پویا تخصیص داده شوند. این رویکرد به کمک الگوریتم رقابت استعماری تضمین می‌کند که هر بار مناسب‌ترین بازه جهت تخصیص منابع پیشنهاد شود که تعداد تخطی را چه در بازه‌های زمانی بلند و چه در بازه‌های زمانی کوتاه کاهش دهد تا از حد مجاز معین‌شده در توافقنامه سطح سرویس بیشتر نشود. نتایج نشان می‌دهد که رویکرد پیشنهادی در آن مقاله توانسته تعادل نسبی بین میانگین بازه‌های بلند و بازه‌های کوتاه برقرار کند و بهتر از الگوریتم ژنتیک عمل نماید. عدم پیاده‌سازی در محیط کلود سیم جزء نقاط ضعف آن مقاله است (ناهید غفوری و فاطمه سعادت جو،۲۰۱۲)

جدول ۱.مقایسه روش‌های پیشین

نویسنده	*سال*	*روش ارائه‌شده در مقاله*	*مزایا و معایب روش*
وود و همکاران	۲۰۰۹	توسعه یک سیستم با فرآیند خودکار نظارت (Sandpiper)	مزایا: امکان فرآیند خودکار نظارت و تشخیص نقاط مهم و تنظیم مجدد ماشین مجازی در زمان لازم و جلوگیری از نقض توافقنامه سطح سرویس. معایب: عدم نگاشت معیارهای سطح پایین به پارامترهای توافقنامه سطح بالا.
راک و همکاران	۲۰۱۳	ارائه نظارت ابری یک برنامه با استفاده از روش mOSAIC	مزایا: جمع‌آوری اطلاعات برای اجرای دستی یا خورکار تنظیم بار ، کاهش/ افزایش تعداد ماشین‌های مجازی یا محاسبه هزینه کل اجرای برنامه‌ها معایب: هزینه در نظر گرفته نمی‌شود و عمومی نیست
شائو و همکاران	۲۰۱۱	ارائه یک ضمانت در عملکرد برنامه‌های ابری براساس نظارت بر برنامه	مزایا: استفاده از فن‌های داده‌کاوی معایب: عدم اندازه گیری فاصله مؤثر

[1] SALMON is the service responsible for monitoring the services QoS

[2] ADA is Agrement Document Analysis

www.SID.ir

وکیولا و همکاران	۲۰۱۲	ارائه یک میان‌افزار برای توسعه و به‌کارگیری از نرم‌افزار در ابرهای خصوصی یا عمومی	مزایا: ادغام منابع محاسباتی سازمانی با منابع ابری عمومی
توردسن و همکاران	۲۰۱۲	ارائه روش مبادله ابری	مزایا: عملکرد بهتر و کاهش هزینه معایب: عدم پشتیبانی از زمان‌بندی پویا
مارو و همکاران	۲۰۱۲	ارائه یک رویکرد برای سازگاری قالب توافقنامه کاربران با توافقنامه‌های عمومی	مزایا: رفع محدودیت قالب توافقنامه استاتیک ، امان تعریف نگاشت بین توافقنامه عمومی و خصوص برای کاربران معایب: تحمیل هزینه به کاربر
امیکارها و همکاران	۲۰۱۲	ارائه معماری زیرساخت تشخیص نقض توافقنامهDesvi	مزایا: حداقل رساندن تعامل با کاربران و پیشگیری از نقض توافقنامه
ساروانان و همکاران	۲۰۱۵	ارائه روش Bee-MMT به‌منظور بالا بردن کیفیت سرویس و راضی کردن مشتریان	مزایا: کاهش مصرف برق معایب: نقض بیشتر توافقنامه سطح سرویس نسبت به روش‌های دیگر
نویسنده	سال	روش ارائه‌شده در مقاله	مزایا و معایب روش
سارا زنگنه و احمد فراهی	۱۳۹۳	ارائه یک مکانیزم هوشمند جهت یافتن بازه‌های زمانی پویا در معماریDesvi	مزایا: نتیجه بهینه بین بازه‌های کوتاه و بلند در معماریDesvi
ناهید غفوری و فاطمه سعادت‌جو	۲۰۱۲	استفاده از الگوریتم رقابت استعماری جهت یافتن بازه‌های پویا در معماریDesvi	مزایا: برقراری تعادل نسبی بین زمانی که میانگین بازه‌ها بلند است و زمانی که بازه‌ها از بین بازه‌های کوتاه انتخاب می‌شوند معایب : عدم پیاده‌سازی در محیط کلود سیم

در تحیق پیش رو با یک رویکرد هوشمند و پویا به وسیله ترکیب الگوریتم رقابت استعماری و **الگوریتم ماهی‌های مصنوعی** سعی شده است بازه‌هایی جهت نظارت و کنترل برای منابع معرفی گردد که در هر شرایطی چه افزایش و کاهش بارکاری منابع و چه تغییر مفاد توافقنامه بهترین بازه‌های زمانی را جهت کم کردن هزینه‌های نظارت و بهتر کردن کشف خطاها معرفی کرده و بررسی بازه‌های زمانی را در هر یک دقیقه انجام دهیم که به‌خوبی به نیازهای کاربران پاسخ داده شود. ضمن اینکه با استفاده از الگوریتم ترکیبی رقابت استعماری و الگوریتم ماهی‌های مصنوعی سعی شده است مشکلات کار (مرضیه رحمانی، فاطمه سعادت جو، ولی درهمی،۱۳۹۵) تا حدودی پوشش داده شود. انتظار داریم الگوریتم پیشنهادی پاسخ قابل‌قبول ارائه نماید.

۶- هدف‌ها

یافتن بهترین بازه زمانی به‌صورت پویا[1] جهت کنترل تعهدات سطح سرویس.۲- بهینه‌سازی تضاد منافع ناشی از هزینه تخطی‌های کشف نشده و هزینه ناشی از سربار سیستم.۳- اختصاص بازه‌های زمانی پویا وابسته به میزان استفاده منابع تا موجب بهینه شدن هزینه‌های نظارت بر عملکرد منابع و نگاشت معیارها شود.۴- فراهم‌سازی رضایت مشتریان و افزایش سطح اعتماد فضاهای ابری

۷- جنبه نوآوری تحقیق

بنا بر دانش حاصل از مطالعات ما، برای به دست آوردن بهترین بازه‌های زمانی در کنترل تعهدات SLA در محاسبات ابری به صورت ایستا انجام‌شده است که این کار باعث وقفه شده و سربار سیستم را بالا می‌برد و اگر زمان افزایش یابد ممکن است خطاها دیده نشوند (امیکارها و همکاران، ۲۰۱۲) اما روش مورد استفاده ما در این تحقیق جدید بوده و تاکنون گزارش نشده است. سؤالات تحقیق میتوان به صورت زیر بیان کنیم:

۱- آیا یافتن بهترین بازه زمانی به‌صورت پویا جهت کنترل ضمانت‌نامه سطح کیفی خدمات از دیگر روش‌ها بهتر است؟

۱- آیا رویکرد مورد استفاده می‌تواند منجر به بهینه‌سازی تضاد منافع ناشی از هزینه تخطی‌های کشف نشده و هزینه ناشی از سربار سیستم شود ؟

۲- آیا اختصاص بازه‌های زمانی پویا وابسته به میزان استفاده منابع، موجب بهینه شدن هزینه‌های نظارت بر عملکرد منابع و نگاشت معیارها می‌شود؟

و سؤال اصلی تحقیق :

[1] Dynamic

www.SID.ir

Archive of SID

۳- آیا کشف بازه های زمانی در کنترل تفاهم‌نامه سطح کیفی خدمات در سرویس‌های رایانش ابری با استفاده از الگوریتم های بهینه سازی به خوبی انجام خواهد شد ؟

و فرضیه‌های تحقیق را میتوان به صورت زیر بیان کنیم.

۱- یافتن بهترین بازه زمانی به‌صورت پویا جهت کنترل ضمانت‌نامه سطح کیفی خدمات از دیگر روش‌ها بهتر است.

۲- رویکرد مورداستفاده می‌تواند منجر به بهینه‌سازی تضاد منافع ناشی از هزینه تخطی‌های کشف نشده و هزینه ناشی از سربار سیستم شود.

۳- اختصاص بازه‌های زمانی پویا وابسته به میزان استفاده منابع، موجب بهینه شدن هزینه‌های نظارت بر عملکرد منابع و نگاشت معیارها می‌شود.

۸- فرمول‌بندی مسئله و تابع هزینه

جهت ارائه یک‌فاصله اندازه‌گیری بهینه، به منظور یافتن تخطی‌های توافقنامه سطح خدمات در زمان اجرا، ۲ عامل زیر را بررسی می‌کنیم:

۱) هزینه اندازه‌گیری

۲) هزینه تخطی‌های توافقنامه سطح خدمات از دست رفته یا همان هزینه تخطی کشف نشده توافقنامه سطح خدمات و روابط متقابل قابل‌قبول میان این دو عامل

با استفاده از این دو عامل و دیگر پارامترها، یک تابع هزینه (C) براساس فاصله اندازه‌گیری بهینه، تعریف می‌کنیم.

$$C = \mu \times C_m + \sum_{\psi \in \{cpu, memory, storage\}} \alpha(\psi) \times C_v$$

که در آن μ تعداد اندازه‌گیریها، C_m هزینه اندازه‌گیری، $\alpha(\psi)$ تعداد تخطی‌های توافقنامه سطح خدمات کشف نشده و C_v هزینه تخطی‌های از دست رفته توافقنامه سطح خدمات است. این تابع هزینه، پایه‌ای برای تحلیل نتایج به دست آمده از آزمایش‌های بعدی ما محسوب می‌شود. سپس نتایج بدست آمده به الگوریتم ماهی‌های مصنوعی ارسال می‌گردد.

۹- تابع هزینه [1]

به منظور پیشنهاد بازه زمانی پویا در اندازه‌گیری برای تشخیص تخطی از توافقنامه سطح خدمات برنامه کاربردی در زمان اجرا، دو عامل تعیین‌کننده به شرح زیر استفاده می‌شود.

۱. هزینه اندازه‌گیری

۲. هزینه تخطی کشف نشده توافقنامه

تضاد منافع قابل‌قبول بین این دو عامل فاصله اندازه‌گیری بهینه را تعریف می‌کند. همانطور که گفته شد با استفاده از این دو عامل و پارامترهای دیگر یک تابع هزینه (C) تعریف شده است که براساس آن می‌توان یک بازه زمانی اندازه‌گیری را تعریف کرد.

$$C = \mu \times Cm + \Sigma_{\varphi \epsilon [CPU, Memory, Storage]} \ \alpha(\varphi) \times CV$$

که در آن μ تعداد اندازه‌گیری‌ها، Cm هزینه اندازه‌گیری، $\alpha(\varphi)$ تعداد تخطی‌های کشف نشده توافقنامه سطح سرویس و CV هزینه از دست رفتن تخطی توافقنامه است. به عنوان مثال هر روز ۲۴ ساعت و به ازای هر ساعت ۶۰ دقیقه، که در مجموع خواهیم داشت:

یعنی هر روز ۱۴۴۰ دقیقه است. ما فرض می‌کنیم که مقدار بازه برحسب دقیقه باشد. این فرض یعنی ما از محاسبه برحسب ثانیه چشم‌پوشی می‌کنیم. چرا که به نظر می‌رسد که بارکاری تحمیل‌شده به سیستم برای بررسی ثانیه به ثانیه تغییرات، زیاد خواهد شد و محاسبات ما برحسب دقیقه منطقی‌تر است. بعد از اجرای الگوریتم رقابت استعماری و ماهی‌های مصنوعی، تمام کشورها براساس مقدار تابع برازشان به صورت نزولی مرتب می‌شوند. براساس سرعت، عمل بهنگام‌سازی کشورها صورت می‌پذیرد و جهت حرکت آنها مشخص می‌شود. شبه کد الگوریتم بکارگرفته شده در این تحقیق در شکل ۲ نمایش داده شده است.

60×24=1440

[1] Cost Function

www.SID.ir

Archive of SID

۱۰-روش پیشنهادی

روش انجام کار این است که در ابتدا می‌بایست برای هرکدام از پارامترهای توافقنامه سطح سرویس، بارکاری را تولید کرد، برنامه به این شکل خواهد بود که داده‌های موجود توسط الگوریتم رقابت استعماری و الگوریتم ماهی‌های مصنوعی که توسط سرویس‌دهنده و سرویس‌گیرنده ارائه شده است مورد بررسی قرار می‌گیرد که این داده‌ها هر ۳ دقیقه یک‌بار به‌صورت ایستا تخطی‌ها را بررسی می‌کند. این روش برای بررسی تخطی در بازه‌های پویا کاربرد ندارد برای اینکه تخطی‌ها را به صورت پویا بررسی کنیم باید بین این ۳ دقیقه یک سری بازه‌های تصادفی تولید کنیم که تخطی‌ها را هر m دقیقه یک‌بار بررسی کند، سپس براساس این بارکاری تابع هزینه توسط الگوریتم رقابت استعماری محاسبه و بررسی می‌شود و نتایج به الگوریتم ماهی‌های مصنوعی ارسال می‌گردد و این الگوریتم باتوجه به توافقنامه سطح سرویس که در نظر گرفته‌ایم بررسی می‌کند که آیا هزینه قابل قبول هست یا خیر؟ درصورتی که هزینه وارده قابل قبول باشد بازه انتخابی بدون تغییر می‌ماند و اگر هزینه وارده زیاد باشد بازه مناسب بعدی انتخاب می‌گردد. هر بار در این رویکرد با تغییر بارکاری منابع و یا توافقنامه بازه‌ها نیز به صورت خودکار تغییرخواهند کرد. در محیط متلب با اجرای برنامه، زمان و تعداد تکرار برنامه برای بدست آوردن جواب بهینه انجام خواهد شد. ما در این پژوهش چند سیستم با تعداد اجراهای مختلف را مورد بررسی قرار می‌دهیم تا بتوانیم الگوریتم را در شرایط مختلف محک بزنیم و نتایج را پس از تحلیل و بررسی نمایش دهیم. این کار بدین جهت انجام می‌گردد که پس از پایان اجرای آن زمان شبیه‌سازی را اندازه‌گیری کند و مشخص شود با اندازه‌گیری چه بازه‌ای زمان شبیه-سازی کمتر و با اندازه‌گیری چه بازه‌ای زمان شبیه‌سازی بیشتر می‌شود. در این تحقیق، داده‌ها را در بازه‌های زمانی ۴ ساعته به صورت تصادفی تولید می‌کنیم. تولید داده‌ها بستگی به بارکاری میزبان[1]؛ شامل بهره‌وری از حافظه، شبکه و پردازنده است

در تابع اصلی بارکاری، میزان بهره‌وری مربوط به حافظه گرفته می‌شود:

```
double a = cloudletList.get(j).getUtilizationModelRam().getUtilization(i)
```

تخطی‌های صورت گرفته در زمان اجرای بارکاری، در متغیری ذخیره می‌شود. از روی این متغیر، تخطی‌های توافقنامه سطح سرویس با الگوریتم رقابت استعماری تنظیم می‌شود و مشخص می‌گردد که بازه زمانی در State فعلی چه مقدار در نظر گرفته شود. هدف از تابع هدف رقابت استعماری کاهش سربار زمانی تحمیل‌شده به سیستم و افزایش کشف تخطی‌های انجام شده در بازه زمانی تعیین شده می‌باشد. جهت ایجاد یک تعادل بین این دو خواسته، الگوریتم رقابت استعماری تعیین می‌کند که کدام بازه زمانی انتخاب شود. بنابراین الگوریتم رقابت استعماری دارای دو متغیر ورودی تخطی‌های توافقنامه سطح سرویس و بازه‌های زمانی می‌باشد. از آنجا که می‌بایست بازه زمانی در کمترین حالت خود و تخطی‌های توافقنامه سطح سرویس باید در بیشترین حالت خود باشند، در این صورت با محاسبه برازش محلی ذرات(کشورها) که از ابتدا تغییراتی تصادفی داشته‌اند، تغییراتی در تابع صورت می-گیرد که متمایل به تصمیم‌گیری در مورد بازه زمانی می‌شود که نه خیلی کم باشد آن‌قدر که هر تخطی توافقنامه سطح سرویس را در نظر بگیرد و زمان پردازش[2] زیاد شود و نه آن‌قدر زیاد باشد که خیلی از تخطی‌ها را پیدا نکند. سپس نتایج بدست آمده به الگوریتم ماهی‌های مصنوعی ارسال و بهترین بازه زمانی جهت استفاده از سرویس های رایانش ابری استخراج می‌گردد.

در این پژوهش با مدل‌سازی منابع در نرم‌افزار متلب و کلود سیم با استفاده از الگوریتم رقابت استعماری و ماهی‌های مصنوعی به حل مسئله کنترل تعهدات توافقنامه سطح سرویس خواهیم پرداخت. روش پیشنهادی در زیر به‌صورت گام‌به‌گام بیان می‌شود.

مرحله ۱) مدل‌سازی پارامترهای منابع(ماشین‌ها) شامل CPU, Storage, Memory ,…

مرحله ۲) تعیین مفاد تعهدات توافقنامه سطح خدمات شامل:

Avaibility, Storage. Incoming bandwidth, outaging bandwidth

مرحله ۳) یک مجموعه اولیه از کشورها (که در این پایان‌نامه مدل کننده منابع می‌باشند) با تعداد N_{con} ایجاد شده و به‌منظور تأمین تعهدات توافقنامه سطح خدمات مقداردهی می‌شوند.

---

[1] Host

[2] Time Process

www.SID.ir

مرحله ۴) تابع هزینه برای هر مستعمره با استفاده از رابطه (۱-۳) محاسبه شده و توان هر مستعمره به‌صورت زیر تنظیم می‌گردد.

$$CP_c = \frac{1}{OF_c}, \quad c = 1 : N_{con}$$

که در آن OF_c مقدار تابع هدف برای کلونی cام می‌باشد.

مرحله ۵) به تعداد امپراطوری‌ها (N_{imp}) از قوی‌ترین مستعمرات به‌عنوان استعمارگر در نظر گرفته‌شده و توان هرکدام از استعمارگران به‌صورت زیر تعریف می‌شود:

$$IP_i = \frac{1}{OF_i}, i = 1 : N_{imp}$$

مرحله ۶) اختصاص مستعمرات به هر استعمارگر طبق IP_i محاسبه‌شده. این بدان معناست که تعداد مستعمرات هر استعمارگر متناسب با توان آن استعمارگر می‌باشد.

$$\frac{IP_i}{\sum_{j=1}^{N_{imp}} IP_j} \times \left(N_c - N_{imp}\right)$$

مرحله ۷) حرکت مستعمرات به سمت استعمارگر مربوطه (کشور مستعمره، به‌اندازه X واحد با زاویه θ در راستای خط واصل مستعمره به استعمارگر مربوطه، حرکت کرده و به موقعیت جدید کشانده می‌شود. در شکل (۱-۳)، فاصله میان استعمارگر و مستعمره با d نشان داده شده‌است و X نیز عددی تصادفی با توزیع یکنواخت می‌باشد. پس برای X داریم:

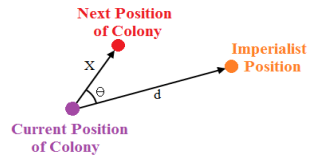
$$X \sim U(0, \beta \times d)$$

که در آن β عددی بزرگ‌تر از ۱ و نزدیک به ۲ می‌باشد. یک انتخاب مناسب برای β می‌تواند عددی بین ۱,۵ تا ۲ فرض شود.

زاویه انحراف θ نیز عددی تصادفی بوده و به‌صورت زیر بیان می‌شود:

$$\theta = (-\gamma, \gamma)$$

که γ زاویه انحراف بوده و محدوده مناسب برای آن می‌تواند زاویه‌ای بین (۴۵،۴۵-) درجه در نظر گرفته شود.)



شکل ۲: حرکت مستعمرات به سمت استعمارگر (سیاست جذب) (آتشپزگرگری و همکاران،۱۳۸۷).

مرحله ۸) انقلاب

مرحله ۹) مقایسه مستعمرات با بهترین موقعیت تجربه‌شده خود و استعمارگر مربوطه. درصورتی‌که هزینه مستعمره کمتر از استعمارگر باشد آنگاه جای این دو باهم عوض می‌شود $(OF_c > OF_i)$. همچنین اگر هزینه مستعمره‌ای کمتر از هزینه بهترین موقعیت تجربه‌شده باشد آنگاه بهترین موقعیت تجربه‌شده بروز می‌شود $(OF_c > OF_{P_best})$.

مرحله ۱۰) محاسبه توان هر امپراتوری بر اساس رابطه زیر:

$$T.C._n = Cost\left(\text{Im}\, perialist_n\right) + \zeta\, mean\left\{Cost(colonies of\ impire_n)\right\} \quad (۳-۶)$$

در این تحقیق مقدار ζ برابر ۰/۲ فرض می‌شود.

www.SID.ir

مرحله ۱۱) ضعیف‌ترین امپراطوری یکی از مستعمرات خود را به‌تصادف از دست می‌دهد. این مستعمره به یکی از امپراطوری های دیگر اضافه می‌شود (احتمال انتخاب امپراطوری براساس توان آن).

مرحله ۱۲) اگر یک امپراطوری بدون مستعمره شد، آن امپراطوری حذف و استعمارگر آن به‌عنوان یک مستعمره به یکی از امپراطوری‌های دیگر اضافه می‌شود.

مرحله ۱۳) ارضا شدن شرط توقف الگوریتم و بازگشت به مرحله ۷.

مرحله ۱۴) ارسال نتایج الگوریتم رقابت استعماری به الگوریتم ماهی‌های مصنوعی.

مرحله ۱۵) پایان.

۱۱- شبه کد تابع هزینه

```
Random i = new Random();
int intrval = i.Next(5,20);
for (int y =0; y < intrval; y++)
{
//monitoring resource and SLA Metrics
//use LoM2HiS to map Metrics
)// at the end of intrval
// evaluates the measurments
int u=0; // number of measurments
int cm =0; // cost of Mesurment
int alfa=0; //undetected SLA violations
int cv=0; // Cost of Undetected SLA violations
int c=0; // total cost
int co=0;
for (int o=0; o<3; o++)//CPU,Memory and Storage
{
    int co = alfa [o]* cv[o];
}
//evaluating the total cost
c = u* cm + co;


}
```

شکل ۲. شبه کد محاسبه تابع هزینه

در این تحقیق برای پیاده‌سازی الگوریتم پیشنهادی تعداد تکرارها و تعداد کشورها (جمعیت) به ترتیب برابر با ۴۰۰ و ۱۰۰ در نظر گرفته شده‌است. همچنین تعداد امپراطوری ها برابر با ۲۰ و احتمال رخ دادن انقلاب در مستعمرات یک امپراطوری نیز روی ۴۰٪ تنظیم شده‌است.

۱۲- تجزیه و تحلیل داده و حجم داده ها SLA

در شرایط مدیریت داده‌ای SLA ما تمامی عملیات داده‌ای SLA را به کار می‌گیریم که قبل و بعد از اجرای خدمات روی می‌دهند. چنین داده‌هایی شامل SLA های فعال از پیش تعریف شده و عناصر خدماتی هستند که ترکیب محتوای SLA را فراهم می‌کنند. در اقتصاد خدمات فرمول بندی SLA از یک ساختاری استفاده می‌کند که فرصت‌های به کارگیری SLA را در برابر عملیات متمرکز بر تجارت گسترش می‌دهد. یک مدل داده‌ای SLA برای حمایت از مدیریت خودکار فرآیندها باتوجه به کنترل و بازرسی داده‌ها موردنیاز است. اشاره به اندازه اطلاعاتی دارد که ذخیره و پردازش می‌شود . در مورد SLA ها و اجزایSLA، اندازه ناچیز است، بنابراین هیچ نیاز مستقیمی برای تخصیص ذخیره یا انتقال داده‌های بزرگ وجود ندارد. حجم در هر سرویس و اندازه داده در هر تعریف ضمانت در دامنه کیلو بایت قرار می‌گیرد. به علاوه، عملیات داده‌ای SLA (به عنوان مثال مشتری یا هشدارهای فراهم کننده) می‌تواند فضای موقتی را به جای ذخیره دیسک اشغال کند. از طرفی دیگر، فرض استفاده از SLA ها توسط مکان‌ها و بازارهای عمومی می‌تواند حجم گسترده‌ای از تقاضاهای مشتریان در داده SLA ذخیره شده باشد. بنابراین یک ساختار داده‌ای انعطاف پذیر و شمای پایگاه داده‌ای باید برای حالت‌های مقاومتی که نیازهای داده‌ای SLA را بررسی می‌کند امکان‌پذیر باشد و پردازش‌های موازی بررسی‌های داده‌ای بلا درنگ را اجازه می‌دهد.

۱۳- پیاده سازی

برای اینکه بتوانیم نتایج را به درستی پیاده‌سازی نماییم و با کارهای انجام شده مقایسه نماییم برنامه را در محیط متلب شبیه‌سازی نمودیم و با استفاده از توابع هزینه که قبلا معرفی نمودیم پژوهش خود را شبیه‌سازی می‌کنیم و در نهایت آزمایش‌های انجام شده را مورد تحلیل و بررسی قرار می‌دهیم. ما برای پیاده‌سازی از برنامه‌نویسی متلب استفاده نمودیم و در یک محیط مجازی پژوهش خود را پیاده‌سازی کردیم در محیط شبیه‌سازی ۵۰ دستگاه فیزیکی طراحی و در هر دستگاه فیزیکی یک ماشین مجازی میزبان شده است. با شروع کار، ماشین‌های مجازی بر روی میزبانان فیزیکی تخصیص داده می‌شوند. درنتیجه باعث ایجاد یک محیط مجازی ابر تا ۵۰ گره

www.SID.ir

محاسباتی می‌شود که قادر به تأمین منابع لازم برای برنامه‌های کاربردی است. در جدول (۱-۴) مشخصات سخت‌افزاری دستگاه‌های فیزیکی و ماشین‌های مجازی طراحی‌شده آورده شده است. ما پژوهش خود را جهت پیاده‌سازی روش پیشنهادی خود روی دو مدل آزمایشی یکی با ۵۰ ماشین مجازی و دیگری با ۱۰۰ ماشین مجازی روی سیستم های کوچک و بزرگ پیاده‌سازی و نتایج را مورد تحلیل و بررسی قرار می‌دهیم. روش کار به این شکل می‌باشد که ماشین‌های مجازی برای تامین سرویس‌های موجود پیاده‌سازی می-شوند و با استفاده از تابع هزینه موقعیت ماشین‌های مجازی مورد ارزیابی قرار می‌گیرد. برای بهینه‌سازی موقعیت ماشین‌ها به‌منظور کم کردن نقص‌های توافقنامه سطح سرویس، از الگوریتم ترکیبی رقابت استعماری و ماهی‌های مصنوعی استفاده می‌نماییم و در پایان تکرارها، بهینه‌ترین موقعیت برای ماشین‌های مجازی، میزان نقص‌های توافقنامه سطح سرویس را تعیین می‌نمایم.
مشخصات فنی ماشین‌های مجازی جهت توافقنامه سطح سرویس در جدول۲ نمایش داده شده است.

جدول۲. مشخصات فنی ماشین‌های مجازی.

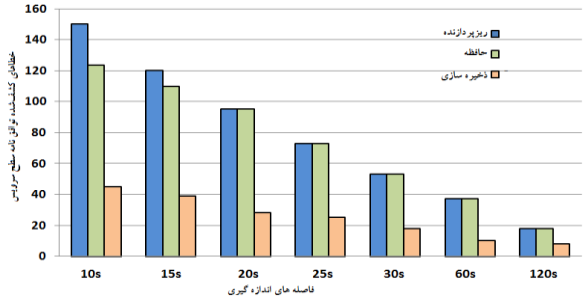
Machine type	*CPU (GHZ)*	*Cores*	*Memory (GB)*	*Storage (GB)*
Physical Machine	GHZ۶	۵	۱۲	۳۰۰
Virtual Machine	GHZ۳	۲	۲	۱۰

تعدادی از تعهدات توافقنامه سطح سرویس که سرویس دهنده‌های خدمات وظیفه دارند آنها را رعایت کنند در جدول ۳ نمایش دادیم.

جدول۳. تعهدات توافق نامه سطح سرویس

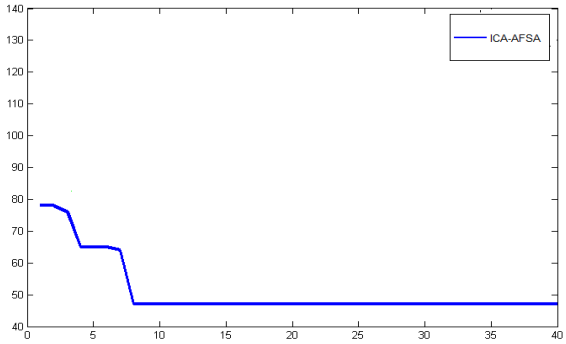
پارامترهای توافق نامه سطح سرویس	
CPU	**۹۵/٪۵**
Memory	**۲/۳۵ GB**
Storag	**۳/۲۰ GB**

اندازه‌گیری توافقنامه سطح خدمات شامل نظارت بر ماشین‌های مجازی ،پردازش داده‌های نظارت شده بررسی معیارهای سطح پایین و بالای توافقنامه سطح سرویس مورد ارزیابی به وسیله تابع هزینه قرار خواهد گرفت.در شکل ۳ نمودار فاصله‌های اندازه‌گیری براساس تعداد تخطی‌های اندازه‌گیری شده نشان داده شده است. و **منحنی سربار اضافی برحسب فاصله اندازه‌گیری** در شکل ۴ نشان داده شده است. همان‌طور که دیده می‌شود با افزایش فاصله اندازه‌گیری‌ها، تعداد خطاهای کشف‌شده نیز افزایش می‌یابد که امری واضح می‌باشد.



شکل ۳: منحنی فاصله اندازه‌گیری براساس خطاهای کشف‌شده

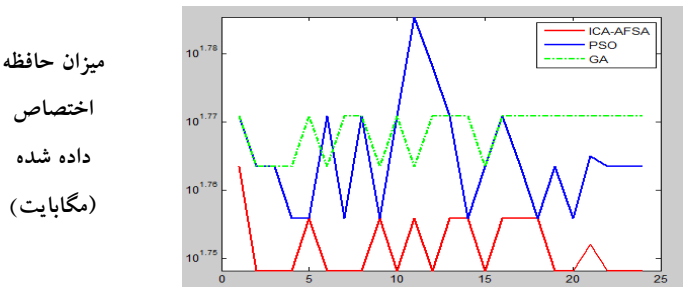
www.SID.ir



شکل ۴: منحنی سربار اضافی برحسب فاصله اندازه‌گیری

۱۴ – تخطی‌های توافقنامه سطح سرویس در الگوریتم رقابت استعماری و ماهی‌های مصنوعی با درخواست کاربر از حافظه

ما بر این فرض هستیم که درخواست کاربر از حافظه حداقل ۸۰۰ مگابایت بوده است. همانطور که در شکل ۵ مشاهده می‌کنید در زمانهای مختلف از روز مقادیر متفاوتی از حافظه توسط سرویس‌دهنده به کاربر اختصاص داده شده است. زمانی که داده‌ها بزرگتر هستند تعداد تخطی‌های کشف نشده بیشتر خواهند بود.

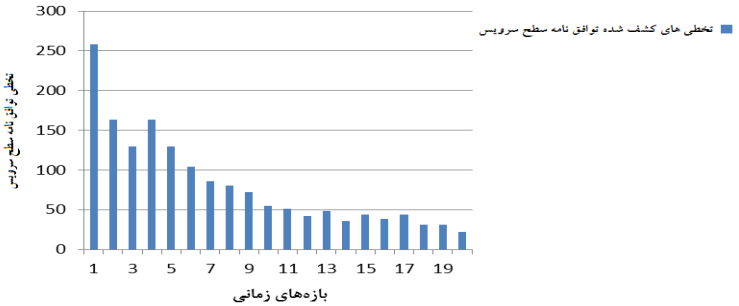


شکل ۵ .میزان حافظه اختصاص شده

۱۵-تخطی‌های توافقنامه سطح سرویس کشف شده برای حافظه در الگوریتم رقابت استعماری و ماهی‌های مصنوعی

همانطور که در شکل ۶ مشاهده می‌کنید به ازای بازه‌های زمانی، تخطی‌های توافقنامه سطح سرویس محاسبه شده است. به عنوان مثال برای بازه زمانی ۷ دقیقه مقدار ۴۶۰ برای حافظه به دست آمده است. این مقادیر قبل از این که به الگوریتم رقابت استعماری و ماهی‌های مصنوعی داده شود به دست آمده است. همانطور که در شکل۴-۵ مشاهده می‌کنیم زمان مصرفی برای کشف کردن این تعداد ۴۶۰تایی توافقنامه سطح سرویس حدوداً برابر ۴۴۶۱۸۸۸ برحسب نانوثانیه می‌باشد. هدف الگوریتم رقابت استعماری و ماهی‌های مصنوعی به دست آوردن بازه زمانی است که در آن تخطی توافقنامه سطح سرویس بیشتر و بار زمانی اعمالی به سیستم کمتر شود. در این حال هر دوی اعداد توافقنامه سطح سرویس و بار زمانی به الگوریتم رقابت استعماری و ماهی‌های مصنوعی داده می‌شوند

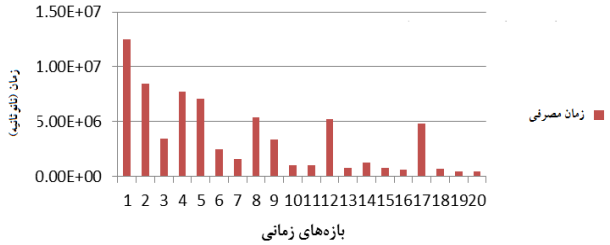
www.SID.ir



شکل ۶ -**نمودار تخطی‌های توافقنامه سطح سرویس کشف‌شده توسط الگوریتم رقابت استعماری و ماهی‌های مصنوعی**

۱۶- سربار اضافه به سیستم در الگوریتم رقابت استعماری و ماهی‌های مصنوعی برای پیدا کردن خطی‌های توافقنامه سطح سرویس

باتوجه به شکل ۷ که در بازه زمانی به طول دو، بازه زمانی دوم، بار زمانی کمتری نسبت به بازه زمانی به طول یک، بازه زمانی اول داریم. لازم است به نمودار تعداد توافقنامه سطح سرویس کشف شده در شکل ۴-۴ توجه کنیم. می‌بینیم که در بازه زمانی اول تعداد توافقنامه سطح سرویس کشف شده بیشتر از بازه زمانی دوم است. کشف این تعداد بیشتر زمان بیشتری را هم بر سیستم تحمیل می-کند. هر چه طول بازه کمتر باشد، تعداد تخطی بیشتری پیدا می‌شود. اما برای این که دفعات بیشتری به بررسی سیستم پرداخته شده است، سربار زمانی تحمیل شده به سیستم بیشتر شده است. همانطور که در شکل ۴-۵ مشاهده می‌کنید، در بازه زمانی اول بیشترین سربار زمانی را داریم. در بازه زمانی سوم تخطی توافقنامه سطح سرویس کمتری پیدا شده بنابراین سربار زمانی کمتری نیز به سیستم تحمیل‌شده است.

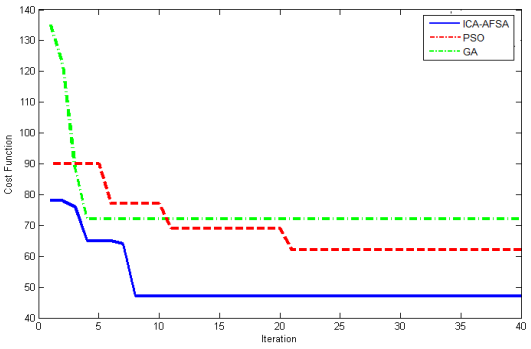


شکل ۷ . **نمودار سربار زمانی تحمیل‌شده به سیستم در ica و ماهی‌های مصنوعی برای پیدا کردن تخطی‌های توافقنامه سطح سرویس**

۱۷- یافتن تخطی توافقنامه سطح سرویس با الگوریتم‌های رقابت استعماری و ماهی‌های مصنوعی، ذرات و ژنتیک

جهت محک زدن کارایی الگوریتم ica و ماهی‌های مصنوعی آزمایش دیگری با داده هایی که به وسیله الگوریتم رقابت استعماری و ماهی‌های مصنوعی پیاده‌سازی شد به وسیله الگوریتم ازدحام ذرات و ژنتیک نیز پیاده‌سازی گردید و به جای استفاده از کشورها در الگوریتم رقابت استعماری از ذره در الگوریتم ازدحام ذرات استفاده شده است و تابه هزینه همان تابع هزینه قبلی خواهد بود همانطور که در شکل ۸ مشاهده می‌کنید الگوریتم ازدحام ذرات در محاسبه بازه زمانی پس از بازه پنجم به بعد نمی‌تواند به‌روزرسانی خود را به خوبی انجام دهد و مقادیر که به دست می آید به سمت بهینگی محلی پیش می‌رود و در بهینگی محلی گرفتار می‌شود. برای یافتن بازه‌هایی که در آن تعادلی بین تعداد توافقنامه سطح سرویس کشف شده و زمان صرف شده برقرار شود، ناتوان می‌ماند و همان مقادیر ابتدایی که برای تشخیص بهترین بازه زمانی را داشته است برمی‌گرداند.

www.SID.ir



شکل ۸ . **نمودار همگرایی تابع هزینه سه الگوریتم رقابت استعماری و ماهی‌های مصنوعی،ذرات و ژنتیک**

www.SID.ir

۱۸- نتیجه گیری

همان‌طور که در بخش های قبل قبل توضیح داده شد برنامه در محیط متلب پیاده‌سازی شده و سیستم شبیه‌سازی شده طبق رویکرد پیشنهادی، مشخصات فنی ماشین‌های مجازی در جدول ۲ و تعهدات SLA در جدول ۳ انجام شد و نتایج بیانگر بهبود روش به‌وسیله الگوریتم رقابت استعماری و ماهی‌های مصنوعی می‌باشد با وقفه‌های کوچک، مقدار سربار افزایش‌یافته و با وقفه‌های اندازه‌گیری بزرگ‌تر مقدار سربار کاهش می‌یابد. همان‌طور که از شکل ۹ پیداست از یک طرف با کاهش تعداد اندازه‌گیرها، هزینه اندازه‌گیری نیز کاهش می‌یابد، و از طرف دیگر هر چه تعداد تخطی‌های SLA کشف‌شده بیشتر باشد، هزینه تخطی‌های ازدست‌رفته نیز بالاتر خواهد بود. این بدان معناست که برای پایین نگه‌داشتن هزینه کشف تخطی‌های SLA، باید تعداد تخطی‌های SLA کشف‌شده نیز پایین باشد.

شکل ۹ **منحنی هزینه اندازه‌گیری و هزینه تخطی‌های توافق نامه سطح خدمات کشف‌شده**

پارامترهای خروجی آزمایش با وقفه کوتاه و بلند به ترتیب در جداول ۴ و ۵آمده است.

جدول۴.پارامترهای خروجی حاصل از آزمایش (وقفه کوتاه)

پارامترها		اولین اجرا	دومین اجرا	سومین اجرا	چهارمین اجرا	میانگین زمان اجرا ICA-AFSA	میانگین زمان اجرا (PSO)	میانگین زمان اجرا (GA)
هزینه اندازه گیری	ICA-AFSA	۵۵,۶	۴۷,۴	۴۳,۱۲	۴۰,۲۶	۴۶,۵۹۵		
	PSO	۵۷,۳۶	۵۹,۲۵	۶۱,۱۴	۵۸,۱۹		58.985	
	GA	۵۹,۴۷	۶۳,۵۵	۶۴,۷۶	۵۸,۹۱			61.6725
تعداد اندازه گیری	ICA-AFSA	۵۴۱۸,۷	۴۷۵۳,۵	۴۴۱۶,۳	۴۴۱۶,۳	4751.2		
	PSO	۵۴۹۸,۶	۵۵۳۳,۶	۵۸۴۷,۷	۵۳۶۶,۱۹		5561.523	
	GA	۵۸۲۵,۲۵	۵۷۴۶,۱۴	۵۹۱۷,۲۳	۶۰۱۲,۲۸			5875.225
SLA نقض غیر قابل تشخیص	ICA-AFSA	۰,۴	۰,۳	۰,۴	۰,۵	0.4		
	PSO	۰,۵	۰,۴	۰,۶	۰,۶		0.525	
	GA	۰,۶	۰,۵	۰,۸	۰,۷			0.65

www.SID.ir

هزینه نقض SLA غیر قابل تشخیص	ICA-AFSA	۸,۲	۹,۳	۷	۸,۴	8.225		
	PSO	۹,۶	۹,۴	۱۰,۲	۹,۷		9.725	
	GA	۹,۹۰	۹,۷	۹,۵	۱۰,۹			9.925

جدول۵. پارامترهای خروجی حاصل از آزمایش (وقفه بلند)

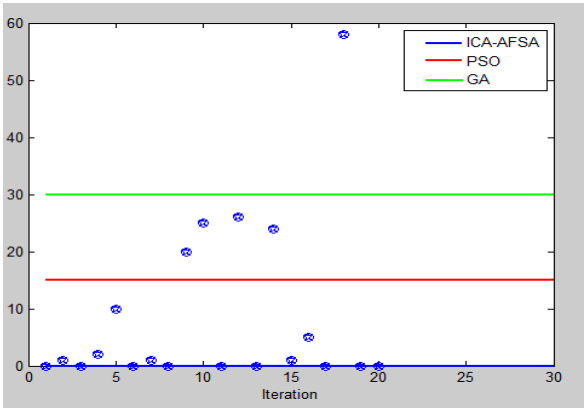
پارامترها		اولین اجرا	دومین اجرا	سومین اجرا	چهارمین اجرا	میانگین زمان اجرا ICA-AFSA	میانگین زمان اجرا (PSO)	میانگین زمان اجرا (GA)
هزینه اندازه گیری	ICA-AFSA	۱۶,۵۶۲۰	۱۴,۵۹۲۰	۱۵,۵۱۱۲	۱۷,۹۷۵۶	16.1602		
	PSO	۱۷,۱۶۵۲	۱۶,۱۴۸۵	۱۸,۲۵۳۴	۱۸,۹۵۸۶		17.63143	
	GA	۱۸,۱۴۲	۱۷,۲۷۸۵	۱۷,۶۶۹۸	۱۸,۶۵۳۲			17.93588
تعداد اندازه گیری	ICA-AFSA	۱۲۰,۲۶	۱۱۶,۱۵۳۱	۱۱۲,۲۲۶	۱۲۵,۴۵	118.5224		
	PSO	۱۲۴,۱۸	۱۱۹,۱۶۲	۱۲۸,۲۵۲۶	۱۲۷,۱۹		124.6962	
	GA	۱۲۷,۱۶	۱۲۹,۱۸۲	۱۲۹,۱۴۵	۱۲۷,۲۵۱۴			128.1846
هزینه نقض SLA غیر قابل تشخیص	ICA-AFSA	۲۵,۸۶۶۱	۲۲,۶۴۲۳	۲۱,۵۷۱۶	۲۵,۱۸۶۰	23.8165		
	PSO	۲۵,۹۶۵۲	۲۶,۱۴۸۵	۲۸,۵۶۱۲	۲۷,۱۶۲۵		26.95935	
	GA	۲۸,۱۴۳۰	۲۷,۵۶۹۶	۲۸,۱۴۷۵	۲۷,۶۶۵۰			27.88128
هزینه نقض SLA غیر قابل	ICA-AFSA	۳۲,۵۵۷۱	۲۹,۳۶۱۴	۳۰,۳۵۱۴	۳۲,۱۴۷۸	31.10443		
	PSO	۳۲,۱۸۲۵	۳۳,۱۶۲۵	۳۴,۲۵۸۶	۳۴,۹۵۶۰		33.6399	
	GA	۳۳,۶۵۸۵	۳۴,۵۸۹۰	۳۵,۶۷۴۱	۳۶,۵۶۸۹			35.12263

۱۹- نمودار مقایسه‌ای الگوریتم رقابت استعماری و ماهی‌های مصنوعی و الگوریتم ازدحام ذرات و ژنتیک

شکل ۱۰ نمودار مقایسه‌ای روش مبتنی بر الگوریتم بهینه‌سازی ترکیبی رقابت استعماری و ماهی‌های مصنوعی و ازدحام ذرات و ژنتیک را نشان می‌دهد. هدف مقایسه سه الگوریتم رقابت استعماری و ازدحام ذرات و ژنتیک یافتن کمترین مقدار درصورت امکان است و اینکه سربار زمانی تحمیل شده به سیستم را کاهش دهند و تعداد تخطی بیشتری پیدا کنند. به عبارتی دیگر در توابع تعریف شده آن‌ها برای همگرایی کافی است تاحد ممکن به صفر نزدیک شوند. همان‌طور که در شکل ۸ ملاحظه می‌شود الگوریتم ازدحام ذرات و ژنتیک با پیشرفت بازه زمانی دیگر توانایی همگرایی خوبی را نداشته، در صورتی که در الگوریتم ترکیبی رقابت استعماری و ماهی‌های مصنوعی از همان ابتدا مقادیر همگرایی خوبی دارند و تقریبا به صفر نزدیک می‌شود و همچنین در نمایش جوابها در شکل ۱۰ این امر به وضوح قابل مشاهده خواهد بود..

www.SID.ir

نتیجه اجرا



شکل ۱۰ **نمودار مقایسه‌ای الگوریتم ترکیبی رقابت استعماری و ماهی‌های مصنوعی و ازدحام ذرات و ژنتیک**

۲۰-نتایج حاصل از تحقیق

باتوجه به نتایج بدست آمده در این پژوهش همانطور که بیان نمودیم هر چه بازه‌های نظارت بر ماشین‌های فیزیکی و مجازی جهت کشف تضاد توافقنامه سطح سرویس کوتاهتر باشد تعداد نقض‌های توافقنامه کمتر کشف خواهند شد و سربار وارد شده به سیستم به دلیل درگیر بودین سیستم افزایش خواهد یافت.

نتایج شبیه‌سازی نشان‌دهنده این است که الگوریتم ترکیبی رقابت استعماری و ماهی‌های مصنوعی توانسته به حالتی بهینه دست یابد و همچنین به صورت تیتر وار نتایج را می‌توان به صورت زیر بیان نمود.

- کشف بازه‌های زمانی ممکن جهت کشف تخطی‌های موجود و از سمتی دیگر باعث افزایش سربار سیستم خواهد گردید.
- یافتن بازه‌های کشف تخطی بصورت پویا با استفاده از الگوریتم ترکیبی رقابت استعماری و ماهی‌های مصنوعی.
- کشف تخطی‌های موجود از توافق‌نامه سطح سرویس با سرعت بالا.
- همگرایی سریع برای بدست آوردن پاسخ با استفاده از الگوریتم ترکیبی رقابت استعماری و ماهی‌های مصنوعی.

۲۱-پیشنهادات

۱- در این کار از الگوریتم رقابت استعماری جهت یافتن بازه بهینه استفاده شده است. می‌توان از الگوریتم‌های بهینه‌سازی دیگری یا ترکیبی از الگوریتم‌های بهینه‌سازی برای انجام این کار نیز استفاده نمود و نتایج آن‌ها را مورد بررسی قرارداد.

۲- یافتن بازه‌های بهینه نظارت بر مصرف منابع جهت نگاشت و یافتن نقض‌های توافقنامه سطح سرویس استفاده شده است. می‌توان این بازه‌ها را جهت نظارت و تقسیم کار منابع مورد ارزیابی قرارداد.

۳- یافتن بازه‌های بهینه نظارت بر مصرف منابع جهت نگاشت و یافتن نقض‌های توافقنامه سطح سرویس استفاده شده است. می‌توان این بازه‌ها را جهت نظارت و تقسیم‌کار وظایف مورد ارزیابی قرارداد.

۴- یافتن بازه‌های بهینه نظارت بر مصرف منابع جهت نگاشت و یافتن نقض‌های توافقنامه سطح سرویس جهت نظارت و تقسیم‌کار منابع.

۵- استفاده از مهاجرت ماشین مجازی در کنترل تعهدات سطح سرویس انجام دهیم که تعداد تخطی ها را کاهش دهد.

بازه‌های پویا جهت بهینه کردن تضاد منافع در تابع هزینه می‌تواند جهت رسیدن به مقادیر بهتر با استفاده از الگوریتم‌های دیگر

www.SID.ir

Archive of SID

بهینه‌تر شود.

۲۲- منابع

آتشپز گرگری, توسعه الگوریتم بهینه‌سازی اجتماعی و بررسی کارایی آن، پایان‌نامه کارشناسی ارشد. دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران., ۱۳۸۷ .

سارا زنگنه و احمد فراهی, رویکرد پیشنهادی یافتن هزینه بهینه کشف خودکار تخطی از توافقنامه سطح خدمات در معماری DeSVi با اختصاص بازه‌های پویا. همایش ملی مهندسی رایانه و مدیریت فناوری اطلاعات، تهران، شرکت علم و صنعت طلوع فرزین, ۱۳۹۳.

مسعود یقینی و محمدرحیم اخوان کاظم‌زاده, الگوریتم‌های بهینه‌سازی فرا ابتکاری. انتشارات:جهاد دانشگاهی (دانشگاه صنعتی امیرکبیر). p. 428 . (شابک:۱-۰۷۸-۲۱۰-۹۶۴-۹۷۸) شهریور ۱۳۹۵.

فهرست منابع غیرفارسی

Emeakaroha, V.C., et al., Towards autonomic detection of SLA violations in Cloud infrastructures. Future Generation Computer Systems, 2012. 28(7): p. 1017-1029.

Goldner, M. and K. Birch, Resource sharing in a cloud computing age. Interlending & Document Supply, 2012. 40(1): p. 4-11.

Ghafuri, N. and F. Saadatjoo, Find the Best Time Intervals in the Control of Service Level Agreement Commitments in Cloud Computing Using Colonial Competitive Algorithm. International Science and Investigation journal, 2015. 4(4): p. 20-35.

Li, J., et al., Online optimization for scheduling preemptable tasks on IaaS cloud systems. Journal of Parallel and Distributed Computing, 2012. 72(5): p. 666-677.

Maurer, M., et al., Cost–benefit analysis of an SLA mapping approach for defining standardized Cloud computing goods. Future Generation Computer Systems, 2012. 28(1): p. 39-47.

Petcu, D., et al., Experiences in building a mOSAIC of clouds. Journal of Cloud Computing: Advances, Systems and Applications, 2013. 2(1): p. 12.

Saravanan, S., V. Venkatachalam, and S.T. Malligai, Optimization of SLA violation in cloud computing using artificial bee colony'. Int. J. Adv. Eng, 2015. 1(3): p. 410-414.

Tordsson, J., et al., Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. Future Generation Computer Systems, 2012. 28(2): p. 358-367.

Vecchiola, C., et al., Deadline-driven provisioning of resources for scientific applications in hybrid clouds with Aneka. Future Generation Computer Systems, 2012. 28(1): p. 58-65.

.Zissis, D. and D. Lekkas, Addressing cloud computing security issues. Future Generation computer systems, 2012. 28(3): p. 583-592.

www.SID.ir