Archive of SID

مقاله پژوهشی

ارائه یک مدل آگاه از انرژی و مبتنی بر زنجیره مارکوف به منظور مدیریت پویای ماشین‌های مجازی در مراکز داده ابری

مهدی رجب‌زاده، ابوالفضل طرقی حقیقت و امیرمسعود رحمانی

چکیده: استفاده از راهکارهای آگاه از انرژی از موضوعات مهم تحقیقاتی در حوزه رایانش ابری است. با کاربرد مؤثر الگوریتم‌های جایگذاری و تجمیع ماشین‌های مجازی، تأمین‌کنندگان ابر قادر خواهند بود مصرف انرژی را کاهش دهند. در این مقاله مدل جدیدی ارائه شده که با بهبود در الگوریتم‌ها و ارائه روش‌های مناسب، به دنبال رسیدن به نتایج مطلوب است. نظارت دوره‌ای بر وضعیت منابع، تحلیل مناسب داده‌های به دست آمده و پیش‌بینی وضعیت بحرانی سرورها به کمک مدل مارکوف پیشنهادی سبب شده است که تا حد امکان از تعداد مهاجرت‌های غیر ضروری کاسته شود. ترکیب الگوریتم‌های ژنتیک و شبیه‌سازی تبرید در بخش جایگزینی در کنار تعریف زنجیره مارکوف جاذب باعث عملکرد بهتر و سریع‌تر الگوریتم پیشنهادی گردیده است. شبیه‌سازی‌های انجام‌شده در سناریوهای مختلف در کلودسیم نشان می‌دهد که در مقایسه با بهترین الگوریتم مورد مقایسه قرار گرفته، در بار کم، متوسط و زیاد، مصرف انرژی کاهش قابل توجهی داشته و این در حالی است که نقض توافقات سطح سرویس‌دهی نیز به طور متوسط ۱۷ درصد کاهش یافته است.

کلیدواژه: **الگوریتم‌های فرااکتشافی، رایانش ابری، زنجیره مارکوف جاذب، کاهش مصرف انرژی.**

۱- مقدمه

ارائه‌دهندگان بزرگ سرویس ابری چندین مگاوات توان مصرفی برای عملیات مرکز داده‌ها مصرف می‌کنند و در سال میلیون‌ها دلار بابت مصرف انرژی الکتریسیته می‌پردازند. انرژی الکتریسیته مصرف‌شده توسط مرکز داده‌ها در سرتاسر دنیا همچنان در حال رشد است. در واقع می‌توان گفت اگر همه مرکز داده‌ها یک کشور بودند، الان این کشور چهارمین کشور مصرف‌کننده انرژی الکتریسیته بود [۱].

هزینه‌های بسیار بالای انرژی در مراکز داده ابری تنها به دلیل تعداد زیاد منابع محاسباتی و عدم کارایی سخت‌افزارها نیست. مطابق شکل ۱، داده‌های به دست آمده از ۵۰۰۰ سرور در طول شش ماه، نشان می‌دهد که سرورها معمولاً بیکار نبوده‌اند ولی از طرفی هم به ندرت بهره‌وری آنها بالا بوده و اکثر اوقات با ۱۰ تا ۵۰ درصد از ظرفیت خود کار می‌کنند. این مسأله باعث افزایش نرخ هزینه به کارایی می‌شود.

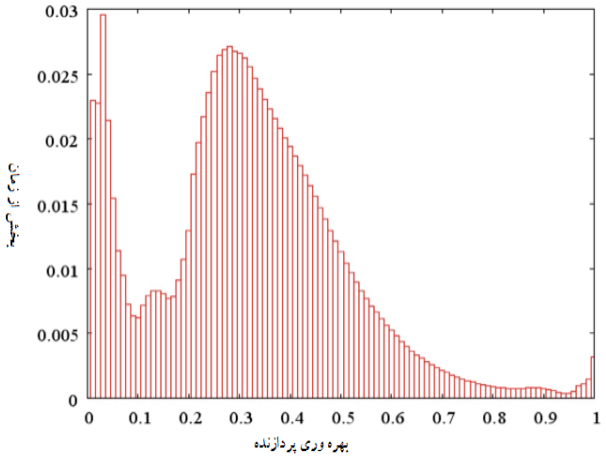
این مقاله در تاریخ ۲۰ دی ماه ۱۳۹۹ دریافت و در تاریخ ۲۷ آبان ماه ۱۴۰۰ بازنگری شد.

مهدی رجب‌زاده، گروه مهندسی کامپیوتر، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران، (email: mehdi.rajabzadeh@srbiau.ac.ir).

ابوالفضل طرقی حقیقت (نویسنده مسئول)، دانشکده مهندسی کامپیوتر، واحد قزوین، دانشگاه آزاد اسلامی، قزوین، ایران، (email: haghighat@qiau.ac.ir).

امیرمسعود رحمانی، واحد علوم و تحقیقات، گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی، تهران، ایران، (email: rahmani@srbiau.ac.ir).

www.SID.ir



شکل ۱: نتیجه بررسی میزان بهره‌وری سرورها طی شش ماه [۱].

از طرفی سرورها در حالت بدون بار یا کم‌بار، بیش از دو سوم زمانی که در اوج بار باشند انرژی مصرف می‌کنند. بنابراین روشن نگه داشتن سرورهای کم‌بار از نظر مصرف انرژی مقرون به صرفه نبوده و باید تا جایی که امکان دارد به کمک تکنیک‌هایی مانند مجازی‌سازی[1] و مهاجرت[2] مجازی از سرورهای کم‌بار و تجمیع آنها در سرورهای دیگر، تعداد کمتری از سرورها را روشن نگه داریم [۲].

در این مقاله برای بهبود فرایند مدیریت منابع و کاهش مصرف انرژی، با شکستن مسأله اصلی به بخش‌های کوچک‌تر، پاسخ‌هایی را برای سوالات زیر ارائه می‌دهیم:

۱) تشخیص میزبان دارای وضعیت بحرانی: وضعیت یک ماشین در چه صورت بحرانی[3] در نظر گرفته می‌شود؟

۲) انتخاب ماشین مجازی: کدام یک از ماشین‌های مجازی باید از میزبان‌ها مهاجرت کنند؟

۳) جایگزینی ماشین مجازی: مقصد ماشین‌های مجازی مهاجرت داده شده، کجا خواهد بود؟

۴) تشخیص میزبان کم‌بار: کدام یک از میزبان‌های کم‌بار، بهترین انتخاب برای خاموش‌شدن به منظور صرفه‌جویی در انرژی هستند؟

یکی از نکاتی که روش پیشنهادی را نسبت به بسیاری از روش‌های ارائه‌شده قبلی متمایز می‌کند این است که در مطالعات قبلی، پرباربودن سرورها، مبنای تصمیم‌گیری برای مهاجرت ماشین‌های مجازی در نظر گرفته شده است. در حالی که از دید ما زمانی باید به آن پرداخت که

1. Virtualization
2. Migration
3. Critical Condition

Archive of SID

احتمال نقض SLA وجود داشته باشد. از این رو با تعریف شرایط بحرانی برای سرورها، بحرانی‌بودن وضعیت را مبنای مهاجرت قرار می‌دهیم. این کار، کاهش تعداد مهاجرت‌ها را در پی داشته و به نوعی از تعداد مهاجرت‌های غیر ضروری می‌کاهد. کاهش تعداد مهاجرت‌ها می‌تواند نقش مهمی در کاهش مصرف انرژی، کاهش سربارهای مهاجرت و افزایش بهره‌وری داشته باشد.

با توجه به توضیحات داده‌شده، به طور خلاصه می‌توان نوآوری‌های پژوهش را در موارد زیر لیست نمود:

۱) ارائه یک مدل جدید آگاه از انرژی
۲) تعریف شرایط بحرانی و اتخاذ تصمیم مربوط به زمان مهاجرت بر مبنای آن
۳) ترکیب زنجیره مارکوف و الگوریتم شبیه‌سازی تبرید به همراه الگوریتم ژنتیک
۴) دسته‌بندی مجازی سرورها و استفاده از مفهوم حالت جذب در زنجیره مارکوف
۵) استفاده از الگوریتم‌های بهبودیافته در سیاست‌های انتخاب و تجمیع ماشین‌های مجازی

در ادامه این مقاله، ضمن بررسی مهم‌ترین پژوهش‌های انجام‌شده در سال‌های گذشته، به تشریح مدل پیشنهادی و الگوریتم‌های به کار رفته در آن می‌پردازیم. در انتها نیز کارایی مدل پیشنهادی به کمک شبیه‌ساز کلودسیم و بر اساس معیارهای مختلف ارزیابی می‌شود.

۲- بررسی کارهای انجام‌شده

در سال‌های اخیر مطالعات متعددی در زمینه جایگذاری و تجمیع ماشین‌های مجازی در مراکز داده ابری صورت گرفته است. هدف بسیاری از این مطالعات، کاهش مصرف انرژی در کنار حفظ کارایی سیستم بوده است [۳] و [۴]. مسأله جایگذاری، معمولاً به عنوان یک مسئله بسته‌بندی چندبعدی در نظر گرفته می‌شود و در این خصوص، چندین الگوریتم با اهدافی متفاوت مانند کمینه‌سازی تعداد سرورهای فعال پیشنهاد شده است [۲] و [۵].

نویسندگان در [۶] تلاش می‌کنند مصرف انرژی را با به کارگیری تجمیع سرور و مکانیزم‌هایی برای تشخیص کم‌باری و پرباری کاهش دهند. همچنین در این تحقیق سعی می‌شود که تعداد مهاجرت را با تعیین کوچک‌ترین مجموعه ماشین‌های مجازی که به منظور مرتفع‌شدن شرایط کم‌باری و پرباری باید مهاجرت داده شوند، به حداقل برسانند.

نویسندگان در [۷] رویکرد جدیدی را پیشنهاد کردند که برای بهبود تجمیع ماشین‌های مجازی از یک مدل مبتنی بر زنجیره مارکوف استفاده می‌کند. آنها با بیشینه‌کردن زمان متوسط بین مهاجرت‌ها به دنبال تحقق اهداف مبتنی بر کیفیت خدمات در کنار کاهش مصرف انرژی هستند.

در [۸] یک چارچوب آگاه از انرژی به منظور نظارت و ارزیابی مصرف انرژی ارائه شده تا به این ترتیب در مدیریت ماشین‌های مجازی بهبود حاصل شود. این چارچوب از شاخص‌هایی مبتنی بر داده‌های بلادرنگ و پیش‌بینی شده از مصرف انرژی که از لایه‌های مختلف شامل لایه برنامه و زیرساخت‌های مجازی و فیزیکی به دست آمده‌اند استفاده می‌کند. این روش مصرف انرژی را به دقت اندازه‌گیری می‌کند. کارایی انرژی بر اساس میزان توان محاسباتی ارائه‌شده به ازای هر وات توان مصرفی محاسبه می‌گردد. با استفاده از شاخص پیشنهادی، میزبان‌هایی که بالاترین کارایی از نظر انرژی را دارند قبل از دیگر میزبان‌ها روشن می‌شوند. ماشین‌های مجازی را می‌توان بر اساس کارایی انرژی کنونی و پیش‌بینی شده در یک میزبان جایگذاری کرد.

در [۹] روشی جهت جایگذاری ماشین مجازی با در نظر گرفتن الزامات پهنای باند و ارتباطات بین ماشین‌های مجازی در برنامه‌های موازی‌سازی شده ارائه گردیده است. این روش، ماشین‌های مجازی دارای ارتباطات متقابل را درون یک سرور جایگذاری می‌کند تا ترافیک شبکه و مصرف انرژی را کاهش دهد. این روش که مهاجرت ماشین‌های مجازی همتا نامیده می‌شود تلاش می‌کند مجموعه ماشین‌های مجازی را که میزبان یک برنامه مشترک هستند، به صورت موازی بر روی یک سرور یکسان مهاجرت دهد و در نتیجه بار شبکه را کاهش دهد. اگر سرور منابع کافی برای میزبانی یک ماشین مجازی نداشته باشد، الگوریتم تلاش می‌کند این ماشین مجازی را حداقل بر روی سروری که بر روی همان لبه شبکه قرار دارد جای دهد. طبق نتایج به دست آمده، استفاده از این روش به طور متوسط موجب کاهش ۲۵٪ در ترافیک شبکه و ۶۰٪ صرفه‌جویی در انرژی می‌شود و به این ترتیب عملکرد ماشین مجازی را به دلیل کاهش تأخیر ارتباطی بهبود می‌دهد.

نویسندگان در [۱۰]، یک معماری سرویس ابری ارائه نمودند که برای پیش‌بینی بار کاری از یک مدل رگرسیون خطی کمک می‌گیرد. در هر دوره زمانی بر اساس بارهای کاری قبلی، سرورهای پربار شناسایی شده و الگوریتم انتخاب ماشین مجازی برای مهاجرت فراخوانی می‌شود.

در [۱۱]، نویسندگان یک مدل آگاه از انرژی و مبتنی بر ابر تحت عنوان مدل زمان‌بندی سبز در ابر ارائه کردند. در مدل پیشنهادی گره‌های ابر، نامتجانس فرض شده و در تخصیص وظیفه و تصمیمات مربوط به زمان‌بندی، قابلیت آگاهی از انرژی در گره‌ها در نظر گرفته شده است. همچنین توانایی گره‌ها در تکمیل وظایف با توجه به محدودیت‌های تعریف‌شده توسط کاربر مد نظر قرار می‌گیرد. به عبارت دیگر مدل پیشنهادی که از آن به GCSM[1] یاد می‌شود، علاوه بر کاهش مصرف انرژی، سعی در برآورده‌کردن کیفیت خدمات مورد نظر کاربر از طریق تکمیل کارهای بلادرنگ در مهلت زمانی آنها دارد.

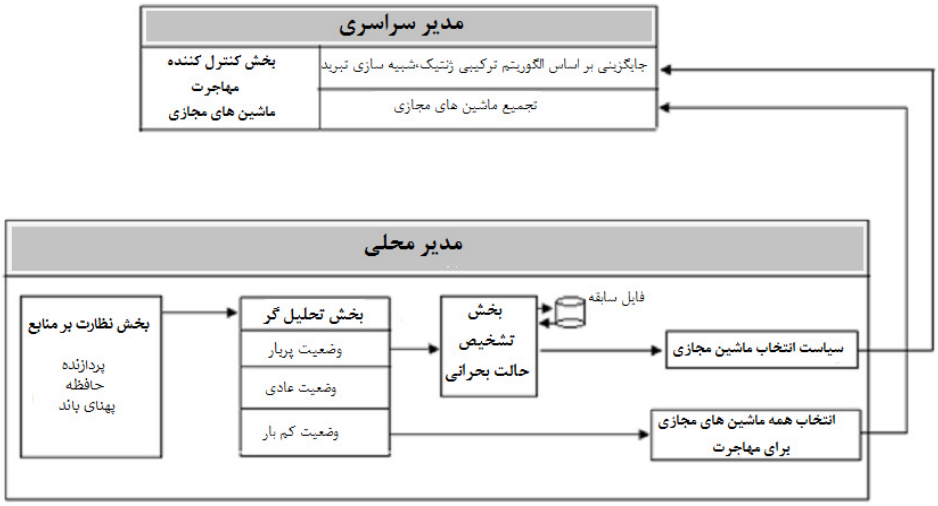
در [۱۲]، یک روش تجمیع مبتنی بر پیش‌بینی به منظور بهینه‌سازی جایگذاری ماشین‌های مجازی بر اساس دو سطح از پیش‌بینی پیشنهاد گردیده است. در این روش، منابع درخواستی ماشین‌های مجازی با استفاده از روش تخمین تراکم هسته پیش‌بینی شده و تخمین ترافیک مهاجرت بین میزبان‌ها در آینده بر اساس سابقه قبلی تخمین زده می‌شود. پیش‌بینی منابع درخواست‌شده در بازه زمانی کوتاه‌مدت آینده امکان کشف وضعیت آینده میزبان را فراهم می‌کند، در حالی که پیش‌بینی ترافیک مهاجرت در آینده، وضعیت هر سرور را برای میزبانی ماشین‌های مجازی جدید فراهم می‌نماید. با توجه به تجمیع پویای ماشین‌های مجازی، از یک چارچوب خاص برای اطمینان از ارتباط بین میزبان‌ها استفاده شده است.

در [۱۳] یک روش تجمیع ماشین‌های مجازی بر مبنای توزیع متوازن بار پیشنهاد گردیده است. ماشین‌های فیزیکی بر اساس ظرفیت منابع به چهار دسته تقسیم شده‌اند تا ماشین‌های مجازی با توجه به تفاوت در درخواست منابع به یکی از این کلاس‌ها نگاشت شوند. طبقه‌بندی کلاس‌ها بر اساس منابع پردازشی، پهنای باند شبکه و حافظه ماشین‌ها انجام شده است. کلاس شماره ۱ بیشترین و کلاس شماره ۴ کمترین میزان منابع را دارا هستند. مهاجرت با هدف کاهش تعداد سرورهای فعال به گونه‌ای انجام می‌شود که هزینه مهاجرت کمینه و بار به طور متوازن بین سرورهای روشن توزیع شود. مهاجرت‌ها تنها بین سرورهای با شماره

1. Green Cloud Scheduling Model

www.SID.ir

Archive of SID



شکل ۲: نمای کلی از مدل پیشنهادی.

کلاس یکسان صورت می‌گیرد و تنها در صورتی که این امر امکان‌پذیر نباشد مهاجرت از سرورهای کلاس بالاتر به کلاس پایین‌تر انجام می‌شود.

در [۱۴]، نویسندگان یک رویکرد آگاه از انرژی و کیفیت خدمات را به منظور چینش و تجمیع چندمنظوره ماشین‌های مجازی پیشنهاد نموده‌اند. هدف تحقیق، دستیابی به یک توازن بین استفاده مؤثر از انرژی، کارایی سیستم و کیفیت خدمات است. رویکرد پیشنهادی شامل دو الگوریتم مکمل است: یک الگوریتم چینش و یک الگوریتم تجمیع چندمنظوره ماشین‌های مجازی به کمک الگوریتم کلونی مورچه. الگوریتم چینش به دنبال یافتن راه حل‌های بهینه جایگذاری ماشین‌های مجازی است که با هدف به حداقل رساندن کل انرژی مصرفی، هدررفت منابع (از نظر پردازنده) و هزینه انرژی ارتباطات (ناشی از بار ترافیک رد و بدل شده بین ماشین‌های مجازی) در یک مرکز داده ارائه گردیده است. با تکیه بر نتایج الگوریتم چینش، الگوریتم تجمیع علاوه بر یافتن راه حل‌های بهینه به منظور به حداقل رساندن مصرف انرژی مرکز داده، به دنبال به حداقل رساندن تعداد مهاجرت‌ها، به حداقل رساندن نقض SLA و به حداقل رساندن تعداد سرورهای فعال است.

در [۱۵] از هزینه مهاجرت ماشین‌های مجازی و زمان باقیمانده از اجرا به عنوان دو عامل مهم که کمتر به آنها توجه شده است نام برده شده است. در الگوریتم پیشنهادی، کل زمان به صورت مجموعه‌ای از قطعه‌های زمانی در نظر گرفته شده و برای اجتناب از مهاجرت‌های غیر ضروری، ماشین‌های مجازی که مقدار باقیمانده از زمان اجرای آنها از یک قطعه زمانی کوتاه‌تر باشد اجازه مهاجرت نخواهند داشت. همچنین هیچ ماشین مجازی نباید بیش از یک قطعه زمانی در حال نقض SLA باشد. هزینه مهاجرت ماشین‌های مجازی نیز به صورت یک رابطه وزن‌دار از نرمال‌شده پارامترهای زمان مهاجرت، زمان غیر فعال بودن و انرژی لازم برای مهاجرت محاسبه می‌شود.

۳- مدل پیشنهادی

در این بخش برای بهبود فرایند مدیریت منابع و کاهش مصرف انرژی، مدلی مطابق شکل ۲ ارائه شده که الگوریتم‌های به کار رفته در آن در ادامه به تفصیل بررسی می‌شوند. در این مدل مدیریت به دو بخش مدیر محلی و مدیر سراسری تقسیم شده که این دو بخش با هم در ارتباط بوده و مدیر محلی در بازه‌های زمانی مشخص، اطلاعات مربوط به وضعیت میزبان را به مدیر سراسری ارسال می‌کند. مدیر سراسری اطلاعات مربوط به همه میزبان‌ها را در اختیار داشته و در تصمیمات مربوط به مهاجرت‌های مورد نیاز برای جایگزینی و تجمیع ماشین‌های مجازی که نیاز به یک دید سراسری از مرکز داده دارد از این اطلاعات استفاده می‌کند.

مدیر محلی روی میزبان‌ها قرار داشته و همان طور که از شکل مشخص است، دارای بخش‌های مختلفی است که در تعامل با یکدیگر عمل می‌نمایند. بخش نظارت بر منابع که وضعیت منابع مانند منابع پردازشی، حافظه و پهنای باند را رصد نموده و در اختیار بخش آنالیز قرار می‌دهد. بخش آنالیز طبق الگوریتم‌هایی وضعیت ماشین‌های محلی را از نظر پربار، معمول یا کم‌بار مشخص می‌کند. در صورت کم‌باربودن، کلیه ماشین‌های مجازی برای مهاجرت انتخاب شده و با هماهنگی با مدیر سراسری اقدامات لازم برای تجمیع صورت می‌گیرد. در صورت پرباربودن نیز وضعیت ماشین از نظر بحرانی‌بودن بررسی شده و در صورت تأیید، طبق سیاست بخش انتخاب ماشین‌های مجازی، موضوع به اطلاع مدیر سراسری رسانده شده و برخی از ماشین‌های مجازی برای مهاجرت انتخاب می‌شوند تا سیستم از حالت بحرانی خارج گردد.

۳-۱ تشخیص وضعیت بحرانی در ماشین‌های فیزیکی

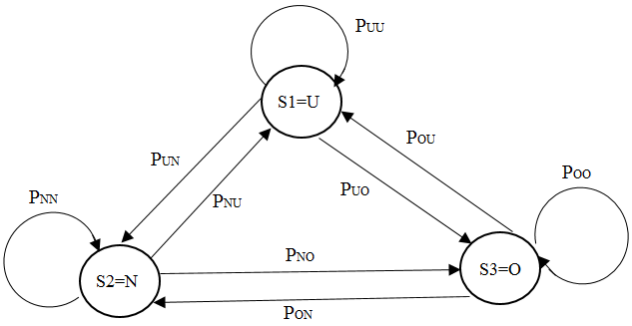
همان طور که قبلاً اشاره شد، هر مهاجرت زنده می‌تواند روی بهره‌وری پردازشی اثر منفی داشته باشد. علاوه بر این در صورتی که سیاست مناسبی در مهاجرت‌ها اتخاذ نشود، ممکن است مجبور شویم برخی از سرورها را از وضعیت خواب به روشن تغییر وضعیت دهیم که هزینه بالایی به لحاظ انرژی در پی دارد. بنابراین در مراکز داده ابری با هزاران میزبان، انجام مهاجرت‌های غیر ضروری تعادل کل سیستم را بر هم می‌زند و تأثیر منفی بر روی کارایی برنامه‌های در حال اجرا خواهد گذاشت [۲]. با توجه به نامتجانس‌بودن سیستم‌ها در مرکز داده، یک میزبان با تعداد هسته‌های بیشتر احتمال کمتری دارد تا با اضافه‌شدن یک ماشین مجازی به حالت پربار برود ولی در همین شرایط یک میزبان با تعداد هسته کمتر احتمال بیشتری دارد تا با اضافه‌شدن این ماشین مجازی به حالت پربار برود. به همین دلیل پرباربودن هر ماشین باید با توجه به شرایط خاص آن ماشین مد نظر قرار گیرد. از طرف دیگر، فقط سطح استفاده از پردازنده مهم نیست و میزان پربودن حافظه اصلی نیز به عنوان یک عامل مهم دیگر باید بررسی شود. ممکن است که یک ماشین از نظر درصد بهره‌وری پردازنده در وضعیت مناسب باشد اما فضای زیادی از حافظه اصلی آن پر شده باشد. در این صورت هم آن ماشین باید به عنوان یک سیستم پربار در نظر گرفته شود زیرا ممکن است منجر به نقض SLA شود.

www.SID.ir

Archive of SID

جدول ۱: تعریف خوشه‌های مجازی بر اساس وضعیت پردازنده و حافظه.

خوشه‌های مجازی	وضعیت پردازنده	تک‌هسته‌ای یا چندهسته‌ای	میزان استفاده از حافظه
C۱	کم‌بار	تک‌هسته‌ای یا چندهسته‌ای	کم
C۲	کم‌بار	تک‌هسته‌ای یا چندهسته‌ای	متوسط
C۳	میان‌بار	تک‌هسته‌ای یا چندهسته‌ای	کم
C۴	میان‌بار	تک‌هسته‌ای یا چندهسته‌ای	متوسط
C۵	کم‌بار	تک‌هسته‌ای یا چندهسته‌ای	زیاد
C۶	میان‌بار	تک‌هسته‌ای یا چندهسته‌ای	زیاد
C۷	پربار	تک‌هسته‌ای	کم
C۸	پربار	چندهسته‌ای	کم
C۹	پربار	تک‌هسته‌ای	متوسط
C۱۰	پربار	چندهسته‌ای	متوسط
C۱۱	پربار	تک‌هسته‌ای یا چندهسته‌ای	زیاد



شکل ۳: مدل پیشنهادی جهت پیش‌بینی وضعیت بحرانی در سرورها.

به همین دلیل در بخش تحلیل‌گر[1]، وضعیت پردازنده و حافظه با هم بررسی می‌شود. با توجه به وضعیت پرباربودن پردازنده و تک‌هسته‌ای یا چندهسته‌ای بودن آن و همچنین میزان فضای اشغال‌شده از حافظه اصلی، حالت‌های یازده‌گانه طبق جدول ۱ تعریف می‌شود.

پرباربودن پردازنده طبق الگوریتم LR[2] سنجیده شده و در صورتی که بار آن زیر ۲۰٪ باشد، کم‌بار در نظر گرفته می‌شود. در مورد حافظه اصلی نیز در صورتی که بیش از ۸۰٪ از فضای حافظه پر باشد، میزان استفاده از حافظه زیاد در نظر گرفته شده و در صورتی که حافظه کمتر از ۲۰٪ اشغال شده باشد، میزان استفاده کم در نظر گرفته شده است. با توجه به جدول ۱ و توضیحات ارائه‌شده، ابتدا خوشه‌ها را بر اساس میزان استفاده از پردازنده و وضعیت حافظه اصلی به ۳ دسته تقسیم می‌کنیم:

- خوشه‌های کم‌بار: سرورهای موجود در خوشه‌های C۱ و C۲
- خوشه‌های معمولی: سرورهایی که در خوشه‌های C۳، C۴، C۵، C۶، C۸ و C۱۰ قرار دارند.
- خوشه‌های پربار: سرورهای موجود درخوشه‌های C۷، C۹ و C۱۱

مدل مبتنی بر مارکوف برای پیش‌بینی وضعیت ماشین‌ها در آینده به سابقه‌ای از وضعیت آنها در دوره‌های قبلی نیازمند است که این داده‌ها در فایل سابقه[3] ذخیره می‌شود. در مدل پیشنهادی، زمانی می‌توانیم پیش‌بینی انجام دهیم که حداقل تعدادی رکورد در این سابقه وجود داشته باشد. در پیاده‌سازی انجام‌شده، حداقل تعداد این رکوردها ۱۰ مورد در نظر گرفته شده است. مدل پیشنهادی یک مدل مارکوف مرتبه اول است که در آن وضعیت آینده بر اساس وضعیت فعلی پیش‌بینی می‌شود [۱۶] و [۱۷]. با توجه به حالت‌های سه‌گانه تعریف‌شده و احتمالات گذر بین آنها، دیاگرام مدل پیشنهادی به صورت شکل ۳ می‌باشد. ماتریس حالات نیز یک ماتریس ۳×۳ به صورت (۱) است

$$P=\begin{pmatrix} P_{UU} & P_{UN} & P_{UO} \\ P_{NU} & P_{NN} & P_{NO} \\ P_{OU} & P_{ON} & P_{OO} \end{pmatrix} \quad (۱)$$

در هر دوره از بررسی، سیستم در یکی از حالت‌های سه‌گانه شکل ۳ خواهد بود. وضعیت بعدی سرورها بر اساس حالت فعلی و احتمالات گذر بین حالت‌ها، محاسبه و پیش‌بینی خواهد شد. بحرانی‌بودن در صورتی تأیید می‌شود که احتمال این که یک سیستم پربار در آینده نیز در حالت پربار باقی بماند از احتمال سایر حالت‌ها بیشتر باشد. به همین ترتیب، مشابه (۲) و با توجه به تعداد تغییر حالت‌ها در فایل سابقه، P_{OU} و P_{ON} نیز محاسبه می‌شود. در صورتی که P_{OO} بیشتر از احتمالات دیگر باشد، یعنی با احتمال بالا وضعیت سیستم در آینده نیز در حالت پرباری بوده و شرایط سیستم بحرانی تعریف می‌شود.

$$P(T_n=O|T_{n-1}=O)=P_{OO}=\frac{P(T_n=O|T_{n-1}=O)}{P(T_n=U|T_{n-1}=O)+P(T_n=N|T_{n-1}=O)+P(T_n=O|T_{n-1}=O)} \quad (۲)$$

۳-۲ انتخاب ماشین مجازی برای مهاجرت

یک مسأله مهم در انتخاب ماشین مجازی برای مهاجرت، زمان مهاجرت است. زمان مهاجرت طبق (۳) به دو عامل میزان استفاده از حافظه اصلی و پهنای باند بستگی دارد. از این رو هرچه میزان استفاده از حافظه اصلی توسط ماشین مجازی کمتر باشد، مهاجرت سریع‌تر انجام می‌شود. هرچه مهاجرت در زمان کوتاه‌تری صورت گیرد هزینه‌های ناشی از آن نیز طبق (۴) کمتر خواهد بود. عامل دیگری که در بسیاری از کارهای انجام‌گردیده کمتر به آن توجه شده است، میزان استفاده ماشین مجازی از پردازنده است. مهاجرتی که در یک زمان کم انجام شده و منجر به آزادشدن درصد بیشتری از پردازنده شود مطلوب‌تر است

$$T_i=\frac{M_i}{B} \quad (۳)$$

$$U_{di}=\cdot/۱\int_{t_\cdot}^{t_\cdot+T_i}U_i(t)\mathrm{d}t \quad (۴)$$

با توجه به نکات ذکرشده، مبنای انتخاب ماشین مجازی برای مهاجرت از سرورهای بحرانی، شاخص مهاجرت است. این شاخص بر اساس نسبت استفاده از پردازنده به حافظه اصلی اشغال‌شده توسط هر ماشین مجازی مشخص می‌شود. الگوریتم این بخش در زیر آمده است:

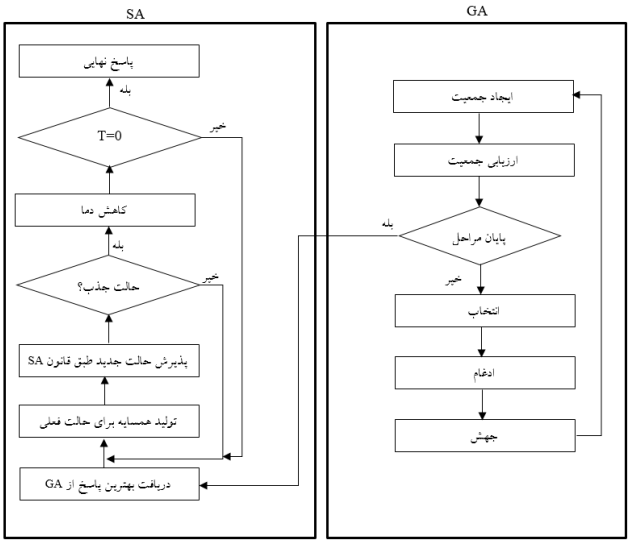
۱) یک لیست از سرورهای بحرانی خوشه‌های C۷، C۹ و C۱۱ تشکیل شود.

۲) یک لیست از ماشین‌های مجازی هر سرور بحرانی تشکیل گردد و به هر یک از آنها یک شاخص مهاجرت نسبت داده شود.

1. Analyzer
2. Local Regression
3. Log

www.SID.ir

Archive of SID



شکل ۴: الگوریتم ترکیبی ژنتیک- شبیه‌سازی تبرید بهبودیافته.

۳) لیست ماشین‌های مجازی به صورت نزولی بر اساس شاخص مهاجرت مرتب شود. اگر برای برخی از ماشین‌های مجازی این پارامتر برابر بود، بر اساس استفاده کمتر از حافظه اصلی مرتب شوند.

۴) اولین ماشین مجازی را از لیست برداشته و در صورتی که منجر به خارج‌شدن سرور از وضعیت بحرانی شود، آن را برای مهاجرت انتخاب نموده و در لیست مهاجرت قرار می‌دهیم. این کار باعث می‌گردد تا سرور از حالت بحرانی خارج شده اما همچنان نزدیک به آستانه بالا بماند.

۵) اگر سرور همچنان در وضعیت بحرانی باشد، ناگزیر به انتخاب ماشین مجازی بعدی از لیست و انتقال آن به لیست مهاجرت هستیم. این کار تا خارج‌شدن سرور از حالت بحرانی ادامه دارد.

۳-۳ سیاست جایگزینی ماشین‌های مجازی

در این قسمت از الگوریتم ترکیبی ژنتیک- شبیه‌سازی تبرید بهبودیافته به منظور جایگزینی ماشین‌های مجازی استفاده می‌شود. الگوریتم ژنتیک می‌تواند با حفظ جمعیت برتر برای نسل بعدی در روند عملکرد ژنتیکی، تنوع جمعیت را تضمین کند. الگوریتم ژنتیک یک الگوریتم بهینه‌سازی سراسری است و الگوریتم شبیه‌سازی تبرید قابلیت جستجوی محلی قوی دارد و قادر به فرار از نتایج بهینه محلی است. در نتیجه، ترکیبی مناسب از این الگوریتم‌ها به گونه‌ای که از مزایای هر دو استفاده شود، می‌تواند منجر به نتایج مطلوب گردد [۱۸] تا [۲۳].

در الگوریتم ترکیبی، ابتدا الگوریتم GA اجرا گردیده و فرد برتر از جمعیت به عنوان ورودی به SA تحویل داده شده و SA فرد بهینه جمعیت GA را بهبود می‌بخشد. از طرف دیگر برای کاهش تعداد تکرارها در حلقه داخلی SA، از مفهوم زنجیره مارکوف جاذب که در ادامه در قسمت ۳-۳-۲ شرح داده شده است استفاده می‌شود، به طوری که تعداد تکرارها بسیار کم می‌گردد. پس از بهبود، الگوریتم بسیار سریع‌تر از GA-SA معمولی عمل می‌کند.

در الگوریتم ترکیبی بهبودیافته، GA جمعیت اولیه را به صورت تصادفی در مرحله اول، تولید و سپس ارزیابی می‌کند و با استفاده از سه عملگر ژنتیک برای تولید جمعیت جدید اقدام می‌نماید. بعد از پایان کار، GA بهترین کروموزوم را برای بهبود بیشتر به SA می‌فرستد (شکل ۴). با این کار، شروع SA به جای یک مقدار تصادفی از یک نقطه خوب خواهد بود و چرخه SA ادامه می‌یابد تا زمانی که شرایط خاتمه الگوریتم محقق شود. در ادامه برخی از مؤلفه‌های مربوط به الگوریتم پیشنهادی با جزئیات توضیح داده می‌شود.

۳-۳-۱ بخش الگوریتم ژنتیک

الگوریتم ژنتیک شامل مراحل تولید جمعیت اولیه، محاسبه تابع هدف برای اعضای جمعیت، انتخاب، ادغام و جهش است که در ادامه هر یک از این مراحل توضیح داده شده است.

جمعیت اولیه

جمعیت اولیه در الگوریتم ژنتیک، مجموعه‌ای از پاسخ‌های پیشنهادی به مسأله است که به هر کدام از این جواب‌ها کروموزوم گفته می‌شود. هر کروموزوم، یک راه حل پیشنهادی برای مسأله است. در مسأله مورد نظر، طول کروموزوم‌ها برابر با تعداد ماشین‌های مجازی و مقدار هر ژن از هر کروموزم برابر با سرور متناظر آن است.

بعد از معرفی ساختار کروموزوم نوبت به تولید جمعیت اولیه می‌رسد. نحوه تولید جمعیت اولیه در ادامه توضیح داده شده است.

تولید جمعیت اولیه

جمعیت اولیه، مجموعه‌ای از کروموزوم‌ها است که باید در شرایط اولیه مسأله صدق کنند. مجموع منابع تخصیص داده شده به ماشین‌های مجازی نباید از کل منابع قابل تخصیص سرور بیشتر باشد.

ارزیابی جمعیت و انتخاب

با توجه به نظریه داروین و برای تولید نسل جدید از جمعیت فعلی، باید کروموزوم‌هایی از این جمعیت را برای ادغام و تکثیر انتخاب کنیم. هنگام اجرای عمل انتخاب، کروموزوم‌هایی که از بهینگی بیشتری برخوردارند (بهینگی هر کروموزوم با استفاده از تابع بهینگی محاسبه می‌شود) شانس بیشتری برای انتخاب خواهند داشت. کروموزوم‌های انتخاب‌شده برای ادغام با دیگر کروموزوم‌ها به منظور تولید نسل جدید در محل دیگری جمع‌آوری می‌شوند. عمل ادغام و تولید کروموزوم‌های جدید در این محل که آن را حوضچه ژنتیک می‌نامیم انجام می‌پذیرد. انتخاب کروموزوم‌ها از جمعیت فعلی نیز به روش‌های گوناگونی انجام می‌پذیرد که ما در روش پیشنهادی از انتخاب مبتنی بر چرخ رولت استفاده می‌کنیم.

اصول کلی انتخاب در این روش بدین صورت است که در گام اول برای هر یک از کروموزوم‌های جمعیت فعلی احتمالی را نسبت می‌دهیم که مقدار احتمال هر کروموزوم، شانس انتخاب آن کروموزوم برای انتقال به حوضچه ژنتیک را نشان می‌دهد. در گام دوم نیز با استفاده از روشی و با توجه به احتمال کروموزوم‌ها، تعدادی از آنها را برای انتقال به حوضچه انتخاب می‌کنیم. در این روش ابتدا کروموزوم‌ها را مرتب کرده و با توجه به میزان بهینگی آنها، رتبه‌ای را به هر یک از کروموزوم‌ها نسبت می‌دهیم. میزان بهینگی هر کروموزوم بر اساس افزایش انرژی مصرفی هر سرور به دست می‌آید. هرچه این افزایش کمتر باشد بهتر است. این مقدار بر اساس (۵) به دست می‌آید و کروموزوم‌ها بر اساس این مقدار بهینگی مرتب می‌شوند. به عنوان مثال بهترین کروموزوم رتبه ۱ و بدترین کروموزوم رتبه N به خود می‌گیرد که N نشان‌دهنده اندازه جمعیت فعلی است

$$F = \frac{\sum_{i=1}^{n} E'_i - \sum_{i=1}^{n} E_i}{\sum_{i=1}^{n} E'_i} \qquad (۵)$$

در این رابطه E انرژی قبل از تخصیص ماشین‌های مجازی و E' انرژی بعد از تخصیص آنها است.

www.SID.ir

Archive of SID

در روش پیشنهادی احتمال مربوط به هر کروموزوم از (۶) به دست می‌آید که در آن $P(i)$ احتمال انتخاب‌شدن کروموزوم i و N تعداد کل کروموزوم‌ها است

$$P(i)=\frac{N-i+1}{\frac{N(N+1)}{2}} \qquad (۶)$$

بعد از به دست آوردن $P(i)$ برای هر کروموزوم، احتمال تجمعی برای هر کروموزم را به دست می‌آوریم که $q(i)$ نشان داده شده و از (۷) به دست می‌آید

$$q(i)=P(i)+q(i-1) \qquad (۷)$$

در این روش پس از آن که احتمال انتخاب کروموزوم‌ها تعیین گردید، یک عدد تصادفی بین ۰ و ۱ تولید می‌شود. سپس کرموزوم‌ها (که به شکل صعودی مرتب شده‌اند) از اول بررسی گردیده و کروموزوم‌هایی که توزیع تجمعی آنها بیشتر یا مساوی عدد تولیدشده باشد، انتخاب می‌گردند.

ادغام

در روش پیشنهادی، کروموزوم‌هایی که در مرحله انتخاب حضور داشتند، با هم ترکیب می‌شوند. روش ترکیبی استفاده‌شده در الگوریتم پیشنهادی، استفاده از ادغام دونقطه‌ای است. این روش دو نقطه متفاوت از کروموزوم را به طور تصادفی انتخاب کرده و سپس ژن‌های بین دو نقطه از دو کروموزم با هم جابه‌جا می‌شوند.

مرحله جهش

در روش پیشنهادی از یک جهش به طور تصادفی استفاده شده است. در واقع یک عدد تصادفی در بازه $[1,N]$، تولید و مقدار ژن متناظر آن با عددی تصادفی در بازه $[1,k]$ جایگزین می‌شود که N تعداد کل ماشین‌های مجازی و k تعداد سرورها است.

بعد از انجام جهش، جمعیت مجدداً طبق رابطه قبل ارزیابی و بهترین کروموزوم به عنوان ورودی به SA داده می‌شود.

۳-۳-۲ بخش الگوریتم شبیه‌سازی تبرید

با توجه به عملکرد شبیه‌سازی تبرید و خروجی‌های تابع F در هر دما و تطبیق آنها با شرایط مسأله که بیانگر مارکوفی‌بودن آن می‌باشد، راهکار زیر ارائه شده است:

۱) برای تشکیل جمعیت اولیه باید چند لیست از سرورها تشکیل شود. این لیست نباید سرورهای پربار، کم‌بار و خاموش را در بر گیرد. برای تهیه این لیست‌ها از سرورهای موجود در خوشه‌های مجازی C۳، C۴، C۵، C۶، C۸ و C۱۰ طبق جدول ۱ استفاده شده است. در این لیست‌ها، سرورها بر اساس میزان استفاده از پردازنده به صورت صعودی مرتب می‌شوند. سپس یک لیست از ماشین‌های مجازی انتخاب‌شده برای مهاجرت تشکیل می‌دهیم (تعداد اعضای لیست سرورها و لیست ماشین‌های مجازی باید برابر باشد و در غیر این صورت سرورهای ابتدایی لیست به علت کم‌بارتربودن تکرار می‌شوند). چند آرایش تصادفی از این لیست سرورها انتخاب و به عنوان جمعیت اولیه در نظر گرفته می‌شود. سپس این جمعیت اولیه (لیست‌ها) مورد ارزیابی قرار می‌گیرند. ملاک ارزیابی نرمال‌شده مجموع افزایش انرژی مصرفی سرورهای لیست طبق (۵) است که هرچه کمتر باشد بهتر است.

۲) تعیین بهترین پاسخ: از بین مجموعه لیست‌ها، بهترین لیست را پیدا کرده و نگهداری می‌کنیم.

۳) تنظیم دمای اولیه $T=T_0$

۴) تکرار مراحل ۵ تا ۸ تا زمانی که به حالت جذب وارد شویم یا تعداد مراحل، کافی به نظر برسد (حلقه داخلی الگوریتم).

۵) برای هر کدام از اعضای جمعیت، تعداد مشخصی همسایه تولید و ارزیابی می‌شوند. تولید همسایه هر لیست با اعمال روش‌های مشخصی انجام می‌شود. برای این کار از روش‌هایی مانند جهش، درج و جابه‌جایی می‌توان کمک گرفت.

۶) همسایه‌های لیست‌ها بر اساس ملاک ارزیابی که نرمال‌شده افزایش انرژی مصرفی است به صورت نزولی مرتب شده و از میان آنها بهترین اعضا، به تعداد جمعیت اصلی، برنده می‌شوند.

۷) هر یک از اعضای جمعیت اصلی با یکی از اعضای جمعیت همسایگان برنده، طبق قانون SA مقایسه می‌شوند (اگر بهتر بود حتماً پذیرفته می‌شود و در غیر این صورت با یک احتمال ممکن است پذیرفته شود).

همان طور که می‌دانیم در الگوریتم شبیه‌سازی تبرید، اگر $\Delta F<0$ باشد، تغییر جدید پذیرفته می‌شود اما اگر $\Delta F>0$ باشد، تغییر جدید بر مبنای الگوریتم متروپلیس پذیرفته و یا رد می‌شود. الگوریتم متروپلیس به این صورت است که یک عدد تصادفی R از توزیع نرمال در بازه ۰ تا ۱ انتخاب شده، در صورتی که $\exp(-\Delta F/T)>R$ باشد، تغییر جدید پذیرفته و در غیر این صورت رد می‌شود.

در واقع با توجه به مطالب ذکرشده، می‌توان گفت که با تکرار تولید جواب‌ها و پذیرش آنها با استفاده از قاعده متروپلیس، یک توالی از جواب‌ها به وجود می‌آید که تشکیل زنجیره مارکوف می‌دهند. پایان یافتن این زنجیره با طول محدود و کافی را می‌توان به رسیدن یک سیستم فیزیکی به تعادل ترمودینامیکی در یک دمای مشخص تشبیه کرد. چون وضعیت سیستم در آینده به حالت‌های قبلی وابسته نیست و تنها به حالت فعلی بستگی دارد در نتیجه شرایط زنجیره مارکوف برقرار است.

در این مسأله ما با توجه به مقادیری که تابع برازندگی F می‌تواند اختیار کند، فضای حالت S را تعریف می‌کنیم. این فضا شامل حالت‌های بیست‌گانه با طول مساوی از بازه (۰،۱) است. از این رو اولین حالت (۰،۰/۹۵) و آخرین حالت تعریف‌شده (۰،۰/۰۵) می‌باشد (زیرا ما به دنبال کاهش مقدار انرژی هستیم). به عبارت دیگر در این مرحله، حالت‌ها در هر دما با یک زنجیره مارکوف مدل گردیده و طول زنجیره مارکوف برابر با تعداد گام‌های تکرار در هر دما است. از آنجایی که در مراحل پایانی حل مسأله که الگوریتم به نقطه مینیمم نزدیک می‌شود، بسیاری از تغییرات پیشنهادی رد می‌شوند و ضریب پذیرش کاهش می‌یابد، در نتیجه طول زنجیره ممکن است به طور نامحدود افزایش یابد. برای جلوگیری از این مشکل نیز، از مفهوم حالت جذب‌کننده در زنجیره مارکوف استفاده شده است.

حالت i را جذب‌کننده می‌نامند اگر با ورود به این حالت، خروج از آن غیر ممکن باشد. در نتیجه حالت i جذب‌کننده است اگر و تنها اگر

$$p_{ii}=1 \text{ and } p_{ij}=0 \text{ for } i\neq j$$

با توجه به ماهیت مسأله، حالت معادل (۰،۰/۰۵) که بهترین حالت است را به عنوان حالت جذب در نظر گرفته و با رسیدن به آن، به تکرار در آن دما خاتمه می‌دهیم. در واقع شرط پایان تکرار در هر دما، رسیدن به حالت جذب‌کننده یا تعداد ثابتی تکرار در آن دما است. البته می‌توان ثابت کرد که این زنجیره مارکوف جاذب، بالاخره به حالت جذب وارد می‌شود ولی برای سرعت بیشتر در پیاده‌سازی الگوریتم در صورتی که پس از تعداد مشخصی تلاش به حالت جذب نرسیم به تکرار در آن دما خاتمه می‌دهیم.

www.SID.ir

Archive of SID

جدول ۲: مشخصات سه نوع ماشین مجازی استفاده‌شده.

نوع ماشین مجازی	تعداد هسته	ظرفیت (MIPS)	حافظه (MB)	دیسک (GB)	پهنای باند (Mb/S)
بزرگ	۱	۲۵۰۰	۱۷۴۰	۲/۵	۱۰۰
متوسط	۱	۲۰۰۰	۸۷۰	۲/۵	۱۰۰
کوچک	۱	۱۰۰۰	۶۱۳	۲/۵	۱۰۰

قضیه: زنجیره مارکوف جاذب تعریف‌شده در نهایت وارد حالت جذب می‌شود.

اثبات: با توجه به ماهیت مسأله و زنجیره مارکوف ایجادشده در هر دما، ارگودیک‌بودن زنجیره واضح است زیرا این احتمال وجود دارد تا از هر حالت به حالت دیگر رسید (نه لزوماً در یک حرکت). فرض می‌کنیم از یک حالت مثل S_i شروع کنیم که i می‌تواند هر یک از حالت‌های ۲۰گانه تعریف‌شده باشد. اگر در حالت (۰،۰/۰۵) باشیم که مسأله حل شده است و در غیر این صورت فرض می‌کنیم که با احتمال P پس از طی n گام از حالت فعلی به حالت (۰،۰/۰۵) برویم. در غیر این صورت با احتمال $P-1$ به حالت‌های دیگر می‌رویم. احتمال این که پس از n بار به حالت جذب نرویم کمتر یا مساوی $(1-P)^n$ است که با افزایش n این احتمال به صفر میل نموده و در نتیجه بالاخره به حالت جذب وارد می‌شویم.

۸) بهترین پاسخ یافته‌شده تا کنون به‌روز می‌شود. اگر این پاسخ در بازه مشخص‌شده به عنوان حالت جذب باشد، به تکرار در این دما خاتمه داده و در صورتی که شرایط خاتمه الگوریتم محقق نشده باشند، دما را طبق (۸) کاهش داده و از مرحله ۴ شروع می‌کنیم

$$T_{K+1} = \alpha \times T_K \qquad (۸)$$

که α پارامتر ثابتی است که مقدار آن به صورت دلخواه از بازه ۰/۸ تا ۰/۹۹ انتخاب می‌شود. بررسی‌های انجام‌شده نشان می‌دهد زمانی که مقدار پارامتر دما بالا است، α می‌تواند کوچک باشد تا سرعت محاسبات افزایش یابد اما با پیشرفت محاسبات و کاهش مقدار پارامتر دما، α باید بزرگ باشد تا همگراشدن روش تضمین شود.

۹) پایان.

۴-۳ انتخاب ماشین‌های کم‌بار

در این مرحله نیز از خوشه‌های تعریف‌شده جدول ۱ در بخش ۳-۱ استفاده گردیده است. ماشین‌های فیزیکی که در خوشه‌های کم‌بار قرار دارند برای خاموش‌شدن در اولویت هستند. به ترتیب، کلیه ماشین‌های مجازی از سرورهای خوشه‌های C۱ و C۲ به سرورهای دیگر خوشه‌ها مهاجرت داده می‌شوند. در واقع در این مهاجرت‌ها چند نکته در نظر گرفته می‌شود. مقصد مهاجرت نباید شامل ماشین‌های خوشه‌های C۷، C۹ و C۱۱ باشد و با پذیرش ماشین مجازی در سرور مقصد، درصد استفاده از پردازنده از ۸۰ بیشتر نشود.

۴- شبیه‌سازی و ارزیابی نتایج

در این مقاله از شبیه‌ساز کلودسیم برای ارزیابی الگوریتم‌ها استفاده شده که آزمایشگاه Cloud Computing and Distribution Systems در دانشگاه ملبورن آن را تهیه کرده است. برای آزمایش‌ها از داده‌های ارائه‌شده پروژه CoMon و زیرساخت نظارت برای PlanetLab استفاده گردید [۱].

جدول ۳: مقادیر پارامترهای اساسی در الگوریتم ترکیبی GA-SA.

الگوریتم ژنتیک	الگوریتم شبیه‌سازی تبرید
اندازه جمعیت اولیه: ۱۰۰ نرخ ادغام: ۰/۶ نرخ جهش: ۰/۶ تعداد مراحل: ۵۰	مقدار اولیه دما: ۱۰۰۰ مقدار پارامتر α: بین ۰/۸ تا ۰/۹۹ تعداد همسایه‌های هر حالت: ۵

۴-۱ مشخصات شبیه‌سازی

در شبیه‌سازی‌ها، پهنای باند سرورها ۱ گیگابیت بر ثانیه تعریف گردیده که نیمی از آن برای مهاجرت و نیمی دیگر برای ارتباطات بین ماشین‌های مجازی در نظر گرفته شده است. دوره‌های زمانی ۳۰۰ ثانیه و مدت شبیه‌سازی ۸۶۴۰۰ ثانیه می‌باشد. شبیه‌سازی‌ها بر روی سیستمی با ۸ گیگابایت حافظه اصلی انجام گردیده و در آنها از سه نوع ماشین مجازی مطابق جدول ۲ استفاده شده است. برای این که نتایج قابل اتکا باشد، از بار کاری واقعی در سناریوها استفاده شده که بخشی از پروژه CoMon است که از بیش از هزار ماشین مجازی از سرورهایی که در بیش از ۵۰۰ نقطه در سراسر جهان قرار دارند، جمع‌آوری شده است. مرجع [۱] در جداول ۲ و ۳ به طور خلاصه اطلاعاتی را از میزان بار کاری مورد استفاده در آزمایش‌ها نمایش داده شده است.

مقادیر جدول ۲ مقادیر پیش‌فرض پیکربندی VM در شبیه‌ساز کلودسیم است. این مقادیر را می‌توان با مراجعه به فایل Constant.java در پوشه Power از شبیه‌ساز کلودسیم تغییر داد و مقادیر جدید را جایگزین کرد. برخی دیگر از پارامترهای شبیه‌سازی نیز در این کلاس مقداردهی شده است. در مورد بار کاری که از Planetlab workload استفاده شده است، این مقادیر نیز در مسیر examples/workload/planetlab در شبیه‌ساز کلودسیم موجود می‌باشند که در این مقاله از مورد ۲۰۱۱۰۳۰۳ استفاده شده است.

توابع مربوط به محاسبات انرژی در کلاس PowerDatacenter.java پیاده‌سازی شده‌اند و برای چاپ در خروجی در کلاس Helper.java به صورت (۹) فراخوانی و محاسبه شده‌اند

$$Energy = \frac{datacenter.\ getpower()}{۳۶۰۰ \times ۱۰۰۰} \qquad (۹)$$

سیاست‌های مربوط به جایگزینی ماشین‌های مجازی روی سرورها در فایل PowerVmAllocationPolicyMigrationAbstract.java که در مسیر org.cloudbus.cloudsim.power می‌باشد، بازنویسی شده است.

۴-۲ معیارهای ارزیابی

برای ارزیابی عملکرد الگوریتم‌ها، از برخی معیارها استفاده شده که در این بخش به طور خلاصه آنها را تعریف می‌کنیم:

OTF: کسری از زمان که طی آن میزبان فعال، استفاده ۱۰۰٪ از CPU را تجربه کرده است.

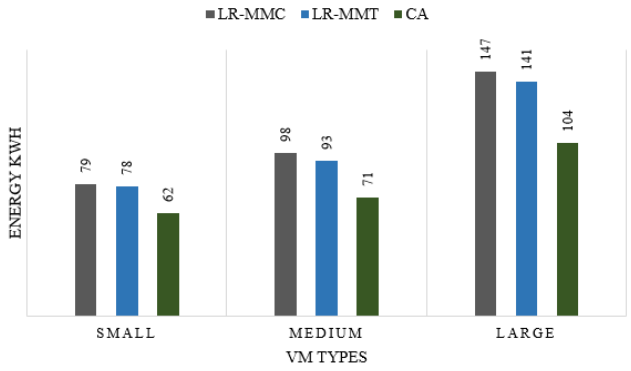
PDM: کل کاهش کارایی که در اثر مهاجرت ماشین‌های مجازی رخ می‌دهد.

معیارهای OTF و PDM برای اندازه‌گیری نقض توافقات سطح سرویس‌دهی مورد استفاده قرار می‌گیرند و مطابق با (۱۰) و (۱۱) محاسبه می‌شوند

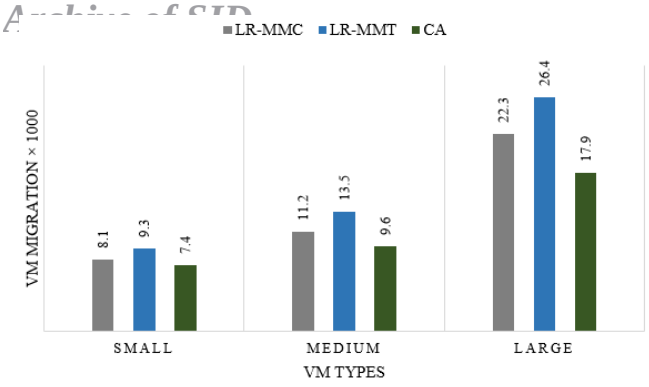
$$OTF = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{ai}}{T_{si}} \qquad (۱۰)$$

www.SID.ir

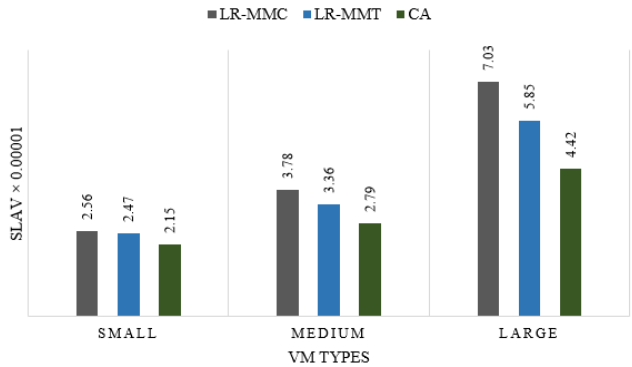
Archive of SID



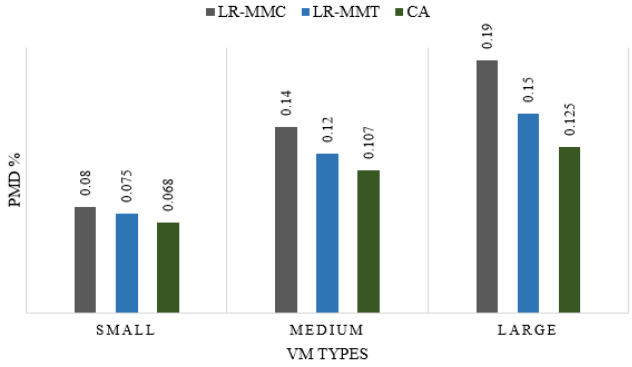
شکل ۵: مقایسه انرژی مصرف‌شده در الگوریتم‌ها.



شکل ۷: مقایسه میزان مهاجرت در الگوریتم‌ها.



شکل ۶: مقایسه نقض SLA در الگوریتم‌ها.



شکل ۸: مقایسه میزان کاهش کارایی در اثر مهاجرت در الگوریتم‌ها.

$$PDM = \frac{1}{M}\sum_{i=1}^{M}\frac{C_{dj}}{C_{rj}} \qquad (11)$$

در (۱۰) و (۱۱)، N تعداد سرورها و T_{si} کل زمانی است که سرور شماره i، استفاده ۱۰۰٪ از پردازنده را تجربه کرده که منجر به نقض SLA شده است. T_{ai} کل سرورهای موجود در حالت فعال، M تعداد VMها، C_{dj} تخمینی از کاهش کارایی ناشی از مهاجرت VM_j و C_{rj} کل ظرفیت پردازنده مورد درخواست VM_j در طول عمر آن است. در این پژوهش، C_{dj} ۱۰٪ از کارایی CPU بر مبنای MIPS در طی کل مهاجرت‌های VM_j تخمین زده می‌شود. هر دو معیار OTF و PDM به طور مستقل، میزان نقض SLA در سیستم را مشخص می‌کنند. بنابراین یک معیار ترکیبی که شامل کاهش کارایی هم در اثر اضافه‌بار میزبان و هم مهاجرت VM است، نشان‌دهنده نقض توافق سطح سرویس بوده که با $SLAV$ معرفی می‌شود. این معیار طبق (۱۲) محاسبه می‌گردد

$$SLAV = OTF \times PDM \qquad (12)$$

$Energy$ کل انرژی مصرف‌شده در مرکز داده و $Migration$ تعداد کل مهاجرت‌هایی است که در n سرور موجود در مرکز داده انجام می‌شود. ESV نیز یک معیار ترکیبی از پارامترهای $Energy$ و $SLAV$ است که طبق (۱۳) محاسبه می‌گردد [۱]

$$ESV = Energy \times SLAV \qquad (13)$$

۴-۳ نتایج شبیه‌سازی و تحلیل آن

پارامترهای اساسی در الگوریتم ژنتیک و شبیه‌سازی تبرید طبق جدول ۳ مقداردهی شده‌اند.

استفاده از مدل مارکوف در پیش‌بینی وضعیت سرورها و به کارگیری الگوریتم ترکیبی GA-SA در بخش جایگزینی ماشین‌های مجازی سبب شده تا مدل پیشنهادی در پارامترهای مبنای مقایسه، عملکرد بهتری را نسبت به الگوریتم‌های دیگر مورد مقایسه داشته باشد. طبق شکل ۵، مجموعه تمهیدات به کار گرفته شده سبب گردیده تا مصرف انرژی در بار کم ۷/۵ درصد، در بار متوسط ۵/۳ درصد و در بار زیاد تا ۷/۱ درصد کاهش یابد.

مجموعه نکات در نظر گرفته شده در مدل پیشنهادی، مانند کاهش مهاجرت‌های غیر ضروری و توجه به وضعیت پردازنده و حافظه از سوی دیگر سبب شده تا نقض SLA، نسبت به الگوریتم‌های مورد مقایسه کاهش داشته باشد. شکل ۶ چگونگی عملکرد الگوریتم‌های مورد مقایسه را در این زمینه نشان می‌دهد. مدل پیشنهادی نسبت به بهترین الگوریتم مورد مقایسه، در بار کم، متوسط و زیاد به ترتیب ۴٪، ۳/۸٪ و ۵/۵٪ کاهش نقض SLA داشته است.

پیش‌بینی وضعیت آینده سرورها و استفاده از خوشه‌های مجازی و راهکارهای به گرفته شده، سبب شده تا از مهاجرت‌های غیر ضروری جلوگیری شود و طبق شکل ۷ تعداد مهاجرت‌های انجام‌شده کاهش یابد. کاهش تعداد مهاجرت‌ها سبب بهبود پارامتر کاهش کارایی ناشی از مهاجرت می‌شود. از آنجایی که تعداد مهاجرت‌ها در مدل پیشنهادی نسبت به سایر موارد کمتر است، در نتیجه مطابق شکل ۸ در زمینه، کاهش کارایی ناشی از مهاجرت، وضعیت بهتری را نسبت به سایر الگوریتم‌ها خواهد داشت.

طبق شکل ۹، مدل ارائه‌شده در زمینه درصد مواقعی که پردازنده در حالت پرباری قرار دارد نیز نسبت به سایر الگوریتم‌ها عملکرد بهتری دارد. با توجه به کاهش مصرف انرژی و کاهش نقض SLA، مدل پیشنهادی نسبت به سایر موارد، در پارامتر ترکیبی ESV نیز طبق شکل ۱۰ عملکرد بهتری خواهد داشت.

۵- نتیجه‌گیری و پیشنهادها

رایانش ابری مزایای بسیار زیادی مانند قابلیت اطمینان، کیفیت

www.SID.ir

Archive of SID

انرژی نیز محقق می‌شود.

دوره به روز رسانی و نحوه رد و بدل کردن اطلاعات بین مدیرهای محلی و مدیر سراسری نیز می‌تواند مورد بررسی و مطالعه بیشتر قرار گیرد. ارائه یک الگوریتم پویا با دقت بالای پیش‌بینی می‌تواند در بهبود کار تأثیر زیادی داشته باشد.

مراجع

[1] A. Beloglazov and R. Buyya, Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing, Ph.D Thesis, Melbourne University, May 2013.

[2] A. Y. Zomaya and Y. C. Lee, *Energy Efficient Distributed Computing Systems*, Wiley-IEEE Computer Society Press, Jul. 2016.

[3] A. Khosravi, S. G. Kumar, and R. Buyya, "Energy and carbon-efficient placement of virtual machines in distributed cloud data centers," in *Proc. of the 19th In. Conf. on Parallel Processing Euro-Par'13,* pp. 317-328, Aachen, Germany, 26-30 Aug. 2013.

[4] J. Yang, C. Liu, and Y. Shang, "A cost-aware auto-scaling approach using the workload prediction in service clouds," *Inf Syst Front*, vol. 16, pp. 7-18, Oct. 2017.

[5] R. Nathuji, C. Isci, and E. Gorbatov, "Exploiting platform heterogeneity for power efficient data centers," in *Proc. of the 4th Int. Conf. on Autonomic Computing*, vol. 7, pp. 5-15, May 2017.

[6] E. Feller, *et al.,* "Energy management in IaaS clouds: a holistic approach," in *Proc. IEEE 5th Inte. Conf. on, Cloud Computing,* pp. 204-212, Honolulu, HI, USA, 24-29 Jun. 2018.

[7] A. Beloglazov and R. Buyya, "Managing overloaded PMs for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints," *IEEE Trans. Parallel Distrib Syst*, vol. 24, no. 7, pp. 1366-1379, Sep. 2013.

[8] G. Katsaros, *et al.*, "A service framework for energy-aware monitoring and VM management in clouds," *Future Generation Computer Systems*, vol. 29, no. 8, pp. 2077-2091, Jan. 2015.

[9] I. Takouna, R. Rojas-Cessa, K. Sachs, and C. Meinel, "Communication-aware and energy-efficient scheduling for parallel applications in virtualized data centers," in *Proc. 6th IEEE/ACM Int. Conf. on Utility and Cloud Computing, UCC'13,* pp. 251-255, Dresden, Germany, 9-12 Dec. 2013.

[10] L. Salimian, F. S. Esfahani, and M. N. Shahraki, "An adaptive fuzzy threshold-based approach for energy and performance efficient consolidation of virtual machines," *Computing*, vol. 98, no. 6, pp. 641-660, Mar. 2016.

[11] K. G. Saurabh, S. Y. Chee, and R. Buyya, "Green cloud framework for improving carbon efficiency of clouds," in *Proc. of the 17th Int. European Conf. on Parallel and Distributed Computing, EuroPar'11,* vol. 6853, pp. 193-203, LNCS, Springer, Germany, Dec. 2011.

[12] T. Mahdhi and H. Mezni, "A prediction-based VM consolidation approach in IaaS cloud data centers," *The J. of Systems & Software*, vol. 146, no. 12, pp. 263-285, Sept. 2018.

[13] I. Mohiuddin and A. Almogren, "Workload aware VM consolidation method in edge/cloud computing for IoT applications," *J. Parallel Distrib. Comput.*, vol. 123, no. 1, pp. 204-214, Sept. 2019.

[14] M. Malekloo, N. Kara, and M. Barachi, "An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments," *Sustainable Computing. Informatics and Systems*, vol. 17, no. 2, pp. 9-24, Feb. 2019.

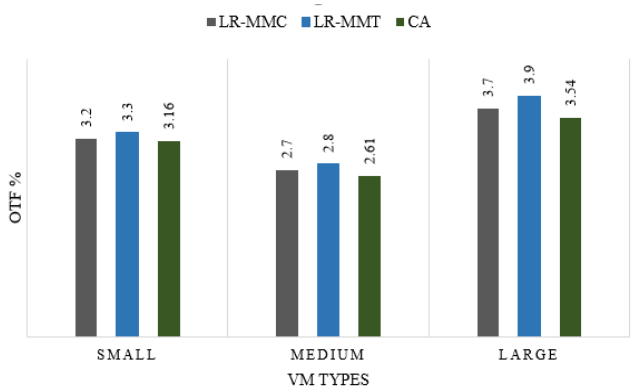
[15] H. Xu, Y. Liu, W. Wei, and Y. Xue, "Migration cost and energy-aware virtual machine consolidation under cloud environments considering remaining runtime," *International J. of Parallel Programming*, vol. 47, no. 1, pp. 481-501 , Mar. 2020.

[16] Z. Luo and Z. Qian, "Burstiness-aware server consolidation via queuing theory approach in a computing cloud," in *Proc. IEEE 27th Int. Symp. on Parallel Distributed Processing, IPDPS'16,* pp. 332-341, Cambridge, MA, USA, 20-24 May 2016.

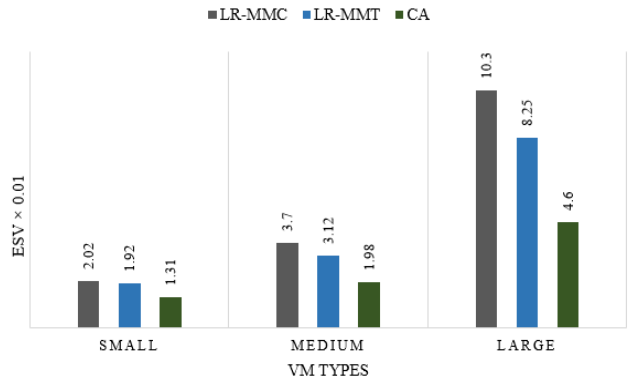
[17] S. B. Melhem, A. Agrawal, N. Goel, and M. Zaman, "Markov prediction model for host load detection and VM placement in live migration, " *IEEE Access*, vol. 6, pp. 7190-7205, 2020.

[18] A. Vasan and K. S. Raju, "Comparative analysis of simulated annealing, simulated quenching and genetic, algorithms for optimal reservoir operation," *Appl Soft Comput.*, vol. 9, no. 1, pp. 274-281, May 2016.

[19] C. Oysu and Z. Bingul, "Application of heuristic and hybrid-GASA algorithms to tool-pathoptimization problem for minimizing airtime during machining," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 3, pp. 389-396, Apr. 2017.



شکل ۹: مقایسه میزان زمان سرریز بار در الگوریتم‌ها.



شکل ۱۰: مقایسه پارامتر ESV در الگوریتم‌ها.

سرویس‌ها و نیرومندی در ارائه آنها را دارا است. از دیدگاه مصرف‌کننده،منابع ابر بی‌نهایت به نظر می‌رسد و مصرف‌کنندگان می‌توانند به همان اندازه که نیاز دارند، قدرت رایانشی خریداری کنند. در مقابل از دیدگاه ارائه‌دهنده ابر، نکته کلیدی بیشینه‌کردن سودآوری و کم‌کردن هزینه‌های عملیاتی است. یکی از مهم‌ترین این هزینه‌ها، هزینه انرژی مصرفی در مراکز داده ابری است. در این پژوهش، یک مدل جامع به منظور کاهش مصرف انرژی با در نظر داشتن توافقات مربوط به سطح سرویس ارائه شده است. در این مقاله برای بهبود فرایند مدیریت منابع و کاهش مصرف انرژی، با شکستن مسأله اصلی به بخش‌های کوچک‌تر، در هر بخش الگوریتم جدیدی را ارائه نمودیم یا الگوریتم‌های قبلی را بهبود دادیم. در مدل پیشنهادی همه مراحل به صورت توزیع‌شده انجام می‌گردد و فقط در جایگزینی ماشین مجازی که نیاز به یک دید سراسری است به صورت متمرکز عمل می‌شود.

به عنوان کارهای آینده، مدل ارائه‌شده در این پژوهش را می‌توان از برخی جنبه‌های دیگر نیز هرچه بیشتر تکمیل نمود. از محورهای نوآوری دیگری که ممکن است در کارهای آینده در روش پیشنهادی مد نظر قرار گیرند می‌توان به موارد زیر اشاره نمود:

طبق تحقیقات انجام‌شده، درصد بالایی از ترافیک مربوط به یک مرکز داده به ترافیک داخلی آن و ارتباطات بین برنامه‌ها برمی‌گردد. در نتیجه استفاده از یک سیاست آگاه از شبکه که سعی می‌کند با بررسی ارتباط بین ماشین‌های مجازی مختلف، آن دسته از ماشین‌های مجازی که ترافیک ارتباطی بین برنامه‌های آنها بیشتر است را روی سرورهای فیزیکی نزدیک به هم جایگذاری نماید، اهمیت خاصی دارد. یک ایده برای پیش‌برد این مسأله، استفاده از خوشه‌بندی و قراردادن ماشین‌های مجازی مرتبط به هم در یک خوشه یا فاصله نزدیک به هم می‌باشد. در این صورت با توجه به کاهش استفاده از تجهیزات شبکه مانند سوئیچ‌ها، کاهش بیشتر در مصرف

www.SID.ir

Archive of SID

[20] M. Rajabzadeh, A. T. Haghighat, and A. M. Rahmani, "New comprehensive model based on virtual clusters and absorbing Markov chains for energy-efficient virtual machine management in cloud computing," *J. Supercomput.*, vol. 76, no. 9, pp. 7438-7457, Dec. 2020.

[21] A. Aryania, H. S. Aghdasi, and L. Mohammad Khanli, "Energy-aware virtual machine consolidation algorithm based on ant colony system," *J. Grid Computing*, vol. 16, no. 2, pp. 477-491, Jul. 2019.

[22] N. Chaurasia, M. Kumar, and R. Chaudhry, "Comprehensive survey on energy-aware server consolidation techniques in cloud computing," *J. Supercomput.*, vol. 77, no. 10, pp. 11682-11737, May 2021.

[23] M. Ala'anzy and M. Othman, "Mapping and consolidation of VMs using locust-inspired algorithms for green cloud computing," *Neural Process Lett.*, vol. 77, no. 10, Oct. 2021.

مهدی رجب‌زاده در سال ۱۳۸۴ مدرک کارشناسی مهندسی کامپیوتر خود را از دانشگاه تربیت معلم تهران و در سال ۱۳۸۷ مدرک کارشناسی ارشد مهندسی فناوری اطلاعات خود را از دانشگاه یزد دریافت نمود. سپس به عنوان عضو هیئت علمی دانشگاه آزاد اسلامی به تدریس در دانشگاه مشغول شد. در سال ۱۳۹۸ موفق به اخذ درجه دکترای تخصصی در مهندسی کامپیوتر از دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران گردید. دکتر رجب زاده اینک استادیار گروه مهندسی کامپیوتر دانشگاه آزاد اسلامی واحد چالوس می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: شبکه‌های کامپیوتری و مخابراتی، رایانش ابری و امنیت داده‌ها.

ابوالفضل طرقی حقیقت در سال های ۱۳۷۱ و ۱۳۷۴ به ترتیب مدرک کارشناسی مهندسی الکترونیک و کارشناسی ارشد مهندسی الکترونیک دیجیتال را از دانشگاه صنعتی شریف دریافت کرد. او در بهمن ۱۳۸۱، موفق به کسب درجه دکترای تخصصی در رشته مهندسی کامپیوتر از دانشگاه صنعتی امیرکبیر تهران شد. از سوابق علمی دکتر حقیقت می‌توان به عضویت در هیأت علمی سازمان انرژی اتمی ایران، عضویت در هیأت علمی دانشگاه آزاد قزوین، ریاست مرکز تحقیقات رباتیک و مکاترونیک و کسب چندین مقام اول جهانی روبوکاپ با تیم های روباتیک اشاره کرد. ایشان هم‌اکنون استادیار دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه آزاد اسلامی واحد قزوین می‌باشد. زمینه‌های علمی مورد علاقه ایشان عبارتند از: شبکه‌های کامپیوتری، سیستم‌های عامل و سیستم‌های توزیعی و رباتیک.

امیرمسعود رحمانی در سال ۱۳۷۵ مدرک کارشناسی مهندسی کامپیوتر خود را از دانشگاه صنعتی امیرکبیر تهران و در سال ۱۳۷۷ مدرک کارشناسی ارشد مهندسی کامپیوتر خود را از دانشگاه صنعتی شریف دریافت نمود. ایشان در سال ۱۳۸۴ موفق به اخذ درجه دکترای تخصصی در مهندسی کامپیوتر از دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران گردید. دکتر رحمانی هم‌اکنون استاد دانشکده مهندسی برق، مکانیک و کامپیوتر دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران می‌باشد. زمینه‌های علمی مورد علاقه ایشان عبارتند از: اینترنت اشیا، یادگیری ماشین، رایانش ابری و کلان داده.

www.SID.ir