

ارائه الگوریتمی برای بهبود جایابی ماشین‌های مجازی در سیستم‌های ابری

فریبا سیمرد^۱، سام جبه داری^۲

^۱ کارشناسی ارشد مهندسی کامپیوتر، دانشکده فنی و مهندسی دانشگاه آزاد اسلامی، واحد تهران شمال.
^۲ دانشیار گروه کامپیوتر، دانشکده برق و کامپیوتر، واحد تهران شمال، دانشگاه آزاد اسلامی، تهران، ایران.

نام نویسنده مسئول:

فریبا سیمرد

چکیده

یکی از چالش‌های اصلی در بحث مهاجرت ماشین‌های مجازی، بحث جایابی ماشین‌های مجازی در محیط ابر می‌باشد. تاکنون روش‌های مختلفی جهت جایابی ماشین‌های مجازی در محیط ابری مطرح شده است که هر کدام، هدف خاصی را دنبال کرده‌اند. در این مقاله به کمک الگوریتم پیشنهادی BBIMO که بر پایه الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی نوشته شده است، جایابی موردی ماشین‌های مجازی را پیشنهاد می‌دهیم. به گونه‌ای که هر ماشین فیزیکی در هر واحد زمانی، حداکثر بتواند یک ماشین مجازی بپذیرد. به این ترتیب هنگام جایابی ماشین‌های مجازی، تداخلی پیش نخواهد آمد و همچنین با مدیریت و کاهش تداخل ماشین‌های مجازی در حال انتقال، افزایش کارایی و بهبود تعادل بار میان ماشین‌های فیزیکی میزبان و کاهش زمان جایابی ماشین‌های مجازی را منجر خواهیم شد.

واژگان کلیدی: جایابی ماشین‌های مجازی- رایانش ابری- مهاجرت- الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی

مقدمه

از آنجاکه استفاده از ماشین‌های مجازی درون مراکز داده و در هر موقعیت مکانی دلخواه به‌تنهایی امکان‌پذیر نیست، باید روشی برای یافتن مناسب‌ترین میزبان فیزیکی برای ماشین‌های مجازی انتخاب شود که این روش را جایابی ماشین‌های مجازی^۱ می‌گویند که در اکثر مراکز داده متناسب با هدف یا شرایط موجود، روش خاصی به‌کاربرده می‌شود.

در این مقاله الگوریتمی بر اساس الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی تحت عنوان BBIMO^۲ برای بهبود جایابی ماشین‌های مجازی در محیط ابری ارائه خواهد شد. به کمک این الگوریتم سعی می‌نماییم تا با کاهش تداخل ماشین‌های مجازی حین انتقال، کاهش زمان اجرا و افزایش تعادل بار را رقم بزنیم.

۱- الگوریتم پیشنهادی

در این بخش الگوریتم پیشنهادی مقاله تحت عنوان الگوریتم BBIMO (الگوریتم بهینه‌سازی و مدیریت تداخل مبتنی بر جغرافیای زیستی) به شرح توضیح داده می‌شود، درواقع این الگوریتم، با استفاده از ویژگی‌های مثبت الگوریتم^۳ BBO و ترکیب آن با الگوریتمی جهت مدیریت و کاهش تداخل ماشین‌های مجازی حین انتقال، به کاهش زمان اجرا و افزایش توازن بار و کارایی محیط ابری می‌پردازد.

الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی یک الگوریتم تکاملی بر پایه جمعیت است که از پدیده مهاجرت حیوانات و پرندگان بین جزایر الهام گرفته شده است. درواقع، جغرافیای زیستی مطالعه توزیع جغرافیایی گونه‌های زیستی می‌باشد.

شاخص میزان مطلوبیت زندگی در هر زیستگاه را با HSI^۴ نمایش می‌دهند. درواقع جزایری که مکان مناسبی برای گونه‌های جغرافیایی جهت اسکان هستند دارای شاخص صلاحیت (HSI) بالا هستند. [۱]

با توجه به الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی، اگر ماشین‌های مجازی به‌عنوان حیوانات و ماشین‌های فیزیکی به‌عنوان زیستگاه‌ها فرض شوند. هر بردار جواب مجموعه‌ای از ماشین‌های فیزیکی خواهد بود که وزن آن‌ها بر اساس تعداد ماشین مجازی جایابی شده در آن تعیین می‌گردد. به این ترتیب با توجه به نحوه جایابی‌های متفاوت ماشین‌های مجازی، بردارهای جواب متفاوتی خواهیم داشت که هر یک میزان شایستگی (HSI) متفاوتی خواهند داشت.

فرض را براین می‌گذاریم که ماشین‌های مجازی موردنظر برای انتقال را انتخاب کرده‌ایم و اکنون زمان انتقال آن‌ها فرارسیده است و ما تنها با چالش جایابی ماشین مجازی (انتخاب مقصد از بین ماشین‌های فیزیکی کاندید)، روبرو هستیم. [۲]

الگوریتم BBIMO شامل ۴ مرحله است که مهم‌ترین مرحله آن استفاده از الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی و تلفیق آن با الگوریتم مدیریت تداخل پیشنهادی است.

با توجه به اینکه هدف ما از بین بردن تداخل ماشین‌های مجازی و درنهایت ایجاد توازن بار است، به شکل رسمی^۵ مسئله‌ی ما به‌صورت معادلات زیر قابل بیان است. در این معادلات متغیر m تعداد ماشین‌های فیزیکی و n نیز تعداد ماشین‌های مجازی است.

i به ماشین فیزیکی شماره i دلالت دارد که در آن $0 < i \leq m$ و j به ماشین مجازی شماره j که در آن $m \leq j \leq n$ دلالت دارد. تابع Res مجموعه منابع اختصاص یافته به هر ماشین فیزیکی و یا مجازی را برمی‌گرداند.

¹ Virtual machine placement

² Biogeography Based Interference Management Optimization

³ Biogeography Based Optimization

⁴ Habitate Suitability Index

⁵ Formal

هدف: کمینه‌سازی عبارت روبروست.

(۱)

$$\sum_{i=1}^m I(\sum_{j=1}^n vm_j x_{ij})$$

هدف:
کمینه‌سازی
عبارت روبرو
با شرط

$$\sum_{j=1}^n Res(vmj).x_{ij} < Res(pmi),$$

$$i \in M = \{1, \dots, m\}$$

$$\sum_{i=1}^m x_{ij} \leq 1, \quad j \in N = \{1, \dots, n\}$$

$$x_{ij} = 0 \text{ or } 1, \quad i \in M, j \in N$$

فرضیات:

$$x_{ij} = \begin{cases} 1 & \text{if } vm_j \text{ is assigned to } pm_i \\ 0 & \text{otherwise} \end{cases}$$

$$Res(vmj) < \max(Res(pmi)), \text{ for } j \in N$$

مجموع تداخل بین ماشین‌های مجازی مستقر بر بسترهای سخت‌افزاری باید کمینه شود. متغیر دودویی X مسأوی یک است اگر ماشین مجازی شماره j به ماشین فیزیکی شماره i منتسب شده باشد در غیر این صورت مساوی با صفر خواهد بود. یک ماشین مجازی فقط به یک ماشین فیزیکی منتسب می‌شود؛ و در نهایت مجموع منابع اختصاص یافته به ماشین‌های مجازی مستقر بر یک ماشین فیزیکی از منابع در اختیار آن ماشین فیزیکی بیشتر نیست و ماشین‌های مجازی باید قابل استقرار بر ماشین‌های فیزیکی باشند. تابع هدف ما برای حل این مسئله، مدیریت و کمینه‌سازی تداخل ماشین‌های مجازی حین انتقال می‌باشد. بطور منطقی واضح است با جایابی موردی ماشین‌های مجازی، (بطوریکه در هر واحد زمانی، هر ماشین فیزیکی موجود در بردار جواب، حداکثر بتواند یک ماشین مجازی جدید بپذیرد و به همین ترتیب هر بردار جواب n سلولی، حداکثر n ورودی بپذیرد)، علاوه بر اینکه تداخل ماشین‌های مجازی حین انتقال را کاملاً از بین می‌بریم، این فرصت را فراهم می‌آوریم که وضعیت ماشین‌های فیزیکی موجود در بردار جواب به لحاظ نرخ مهاجرت پذیری و مهاجرت گریزی چک شوند. به این ترتیب از آنجاکه ماشین‌های مجازی تنها می‌توانند از ماشین‌های فیزیکی پر بارتر به کم بارتر منتقل شوند، در بردار جواب نهایی، علاوه بر از بین بردن تداخل ماشین‌های مجازی، شاهد افزایش تعادل بار بین ماشین‌های فیزیکی موجود در بردار جواب خواهیم بود.

۲- مراحل الگوریتم جایابی ماشین‌های مجازی به کمک الگوریتم BBIMO

به‌طور کلی الگوریتم بهینه‌سازی و مدیریت تداخل مبتنی بر جغرافیای زیستی برای حل مسئله جایابی ماشین‌های مجازی به صورت زیر بیان می‌شود:

۱. مقداردهی اولیه پارامترها (تعیین تعداد ماشین فیزیکی میزبان و تعداد ماشین مجازی ...)
۲. تولید بردار جواب‌ها به طول پارامتر جمعیت
۳. تولید جواب‌های اولیه برای بردار جواب‌ها
- شرح ایجاد راه‌حل‌های اولیه (کروموزوم‌ها) که معمولاً به صورت تصادفی ایجاد شده و در نهایت بهبود می‌یابند، در بخش ۴ به شرح توضیح داده خواهد شد.
۴. محاسبه شایستگی جواب‌ها
۵. مرتب کردن جواب‌ها بر اساس شایستگی آن‌ها به صورت نزولی
۶. انتخاب نیمی از بردارهای جواب از بالا به پایین (بر اساس بیشترین شایستگی) و فرستادن به نسل بعدی
۷. تولید نیم دیگر نسل جدید به کمک نیمه بهینه ارسال شده از نسل قبلی و از بین بردن تداخل ماشین‌های مجازی در حال انتقال به کمک ارسال موردی

باشناسایی پربارترین ماشین فیزیکی و کم‌بارترین ماشین فیزیکی، در هر واحد زمانی یک ماشین مجازی را از پربارترین میزبان به کم‌بارترین میزبان انتقال می‌دهیم تا نیم دیگر نسل جدید تولید گردد.

۸. به کمک نرخ مهاجرت‌پذیری و مهاجرت‌گریزی هر بوم غیرنخبه را اصلاح کرده، عملگر مهاجرت و HSI را دوباره محاسبه می‌کنیم.
۹. برای هر ماشین فیزیکی، احتمال اینکه تعداد ماشین‌های مجازی ساکن در آن را تغییر دهیم وجود دارد. پس هر ماشین فیزیکی غیر نخبه را جهش می‌دهیم، عملگر جهش و سپس مقدار HSI را برای هر بردار جواب دوباره محاسبه می‌کنیم.
۱۰. محاسبه شایستگی جواب‌های جدید
۱۱. برگرداندن بهترین جواب بر اساس شایستگی

۳- تعریف فرض‌های مسئله و تولید راه‌حل‌های تصادفی اولیه (کروموزوم‌ها)

هدف اصلی این است که چالش جایابی ماشین مجازی در محیط ابری را به کمک الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی به گونه‌ای بهبود بخشیم که با مدیریت تداخل ماشین‌های مجازی در حال انتقال به کمک الگوریتم BBIMO، ضمن از بین بردن تداخل میان ماشین‌های مجازی در حال انتقال و ایجاد توازن بار بین ماشین‌های فیزیکی میزبان، باعث استفاده مؤثر از منابع شده و با ایجاد توازن بار افزایش کارایی و کاهش هدر رفت انرژی را رقم بزنیم.

فرض‌های مسئله:

۱. قصد داریم که n ماشین مجازی $VM = \{v1, v2, \dots, vn\}$ را بروی m ماشین فیزیکی $PM = \{P1, P2, \dots, Pm\}$ جایابی نماییم.
 ۲. کروموزوم‌ها (بردارهای جواب) مجموعه‌ای از ماشین‌های فیزیکی هستند که هر کدام وزن خاصی دارند.
 ۳. وزن هر ماشین فیزیکی، نشان‌دهنده جمعیت آن ماشین فیزیکی (تعداد ماشین مجازی تحت میزبانی) می‌باشد.
 ۴. نرخ مهاجرت‌پذیری و مهاجرت‌گریزی ماشین‌های فیزیکی بر اساس جمعیت موجود در آن‌ها ارزیابی می‌شود، بطوریکه هر چه جمعیت یک ماشین فیزیکی بیشتر باشد، نرخ مهاجرت‌پذیری آن کمتر و نرخ مهاجرت‌گریزی آن بیشتر خواهد بود.
 ۵. برای ایجاد تعادل در محیط ابری می‌توان ویژگی‌های ایجاد شایستگی را از یک ماشین فیزیکی پربار، به ماشین فیزیکی کم‌بارتر انتقال داد تا به طبع ایجاد شایستگی در آن ماشین فیزیکی، مهاجرت‌پذیری آن نیز بالا رود و توازن بیشتری در محیط ابری ایجاد شود؛ که این همان معنای Immigration و Emigration در الگوریتم BBO می‌باشد که منجر به تعیین نرخ مهاجرت‌پذیری (λ) و نرخ مهاجرت‌گریزی (μ) برای هر ماشین فیزیکی می‌گردد.
- بنابراین ویژگی‌هایی که باعث ایجاد شایستگی ماشین‌های فیزیکی می‌شوند (SIV)، مانند ظرفیت حافظه، قدرت پردازنده و... از ماشین فیزیکی پربارتر به ماشین‌های کم‌بارتر انتقال می‌دهیم تا با این کار علاوه بر ایجاد توازن شایستگی‌ها، باعث ایجاد توازن بار در ماشین‌های فیزیکی گردیم.

۴- جایابی ماشین‌های مجازی به کمک الگوریتم BBIMO

۴-۱ مرحله اول؛ تعریف جمعیت اولیه و مقداردهی اولیه پارامترها

در این مرحله پارامترهای اصلی مربوط به مسئله مشخص شده و مقداردهی می‌شوند، به‌عنوان مثال در این مرحله مشخص می‌کنیم که چه تعداد ماشین مجازی را قرار است بر روی چه تعداد ماشین فیزیکی جانمایی کنیم، همچنین مقادیر مربوط به نرخ مهاجرت‌پذیری، درصد جهش، تعداد تکرار الگوریتم برای یافتن بهترین جواب و حد بالا و پایین متغیرها را تعیین می‌نماییم. شبه کد مربوط به تنظیمات پارامترهای اولیه مسئله در زیر آمده است.

```
number_VM = 30;
Number_PM = 10;
Max_Iteration = 100;
npop=30;
Nold=ceil(0.5*npop);
Nnew=npop-Nold;
```

(۲)

۴-۲ مرحله دوم؛ تعیین میزان شایستگی ماشین‌های فیزیکی به کمک تابع شایستگی

در این مرحله میزان شایستگی ماشین‌های فیزیکی برای جانمایی ماشین‌های مجازی به کمک تابع Fitness، تعیین می‌گردد. در این تابع بر اساس ۴ آیتم CPU، RAM، Band Width و Storage برای هر ماشین فیزیکی یک‌میزان شایستگی در نظر گرفته می‌شود. تا به کمک آن نرخ

مهاجرت‌پذیری و مهاجرت‌گریزی ماشین‌های فیزیکی موجود در بردار جواب تعیین گردد. هر ماشین فیزیکی با توجه به میزان شایستگی خود، یک نرخ مهاجرت‌پذیری (λ) و یک نرخ مهاجرت‌گریزی (μ) خواهد داشت [۳].

λ & μ	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}
	PM ₁	PM ₂	PM ₃	PM ₄	PM ₅	PM ₆	PM ₇	PM ₈	PM ₉	PM ₁₀
Rates	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}

شکل ۱- نرخ مهاجرت‌پذیری و مهاجرت‌گریزی ماشین‌های فیزیکی

مطابق الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی، نرخ مهاجرت‌پذیری و مهاجرت‌گریزی هر ماشین فیزیکی را به کمک فرمول‌های زیر محاسبه می‌نمائیم [۳].

$$\lambda_i = I(1 - \frac{k(i)}{n}) \quad (3)$$

$$\mu_i = E(\frac{k(i)}{n}) \quad (4)$$

۳-۴ مرحله سوم؛ تعیین شایستگی بردارهای جواب و مرتب‌سازی نزولی آن‌ها بر اساس میزان شایستگی

در این مرحله جمعیت ماشین‌های فیزیکی به صورت تصادفی جایگزین می‌شوند. به این ترتیب بردارهای جواب متعددی که شامل فهرستی از ماشین‌های فیزیکی است ایجاد می‌گردد. در این مرحله میزان شایستگی هر بردار جواب (HSI) ارزیابی می‌شود، سپس بردارهای جواب را بر اساس میزان شایستگی، از بزرگ به کوچک مرتب می‌نمائیم.

۴-۴ مرحله چهارم؛ از بین بردن تداخل ماشین‌های مجازی و ایجاد تعادل بار به کمک الگوریتم BBIMO

در این مرحله ابتدا به کمک الگوریتم BBO جایابی اولیه را به صورت کاملاً تصادفی ایجاد می‌نمائیم. اگر مطابق فرض مسئله M ماشین فیزیکی داشته باشیم، یک ماتریس M^*M ایجاد می‌نمائیم. بر این اساس اگر هر سطر ماتریس را یک بردار جواب در نظر بگیریم در نهایت M بردار جواب خواهیم داشت. (به تعداد سطور ماتریس)

در هر سلول ماتریس عددی بین ۱ تا N قرار می‌گیرد که نشان‌دهنده تعداد ماشین مجازی است که روی آن ماشین فیزیکی قرار گرفته است. به عنوان مثال می‌خواهیم ۳۰ ماشین مجازی را بر روی ۱۰ ماشین فیزیکی به کمک الگوریتم BBIMO جایابی نمائیم. به این ترتیب یک ماتریس 10×30 به صورت زیر خواهیم داشت؛ که در هر سلول آن عددی بین ۱ تا ۳۰ قرار دارد؛ که نشان‌دهنده تعداد ماشین مجازی‌ای است که در آن ماشین فیزیکی جایابی شده است.

PM LIST	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉	P ₁₀
1	7	3	5	2	2	3	1	2	3	2
2	6	4	3	4	2	4	1	2	1	3
3	3	3	5	4	2	3	3	2	3	2
4	3	4	5	2	3	3	1	2	4	3
-	-	-	-	-	-	-	-	-	-	-
10	2	1	5	2	3	4	1	2	5	5

شکل ۲- بردارهای جواب بر اساس تعداد ماشین مجازی

در مرحله بعد برای هر بردار جواب یک شایستگی تعیین می‌نمائیم و ماتریس جواب را از بالا به پایین بر اساس میزان شایستگی جواب مرتب می‌نمائیم. سپس بر اساس الگوریتم BBO دقیقاً نصف این بردار را از بالا انتخاب کرده (شایستگی بیشتر) به نسل بعدی می‌فرستیم و نصف دیگر را دور می‌ریزیم. در مرحله بعد باید نسل جدید را به کمک نسل قبلی و الگوریتم BBIMO ایجاد کنیم. به این ترتیب که ۵۰ درصد جمعیت جدید را از بهترین جواب‌های جمعیت قبلی می‌آوریم و ۵۰ درصد باقیمانده را به کمک الگوریتم BBIMO ایجاد می‌نمائیم.

۴-۵-۵-۴ **طریقه ایجاد ۵۰ درصد جمعیت به کمک الگوریتم BBIMO:**

در این قسمت ابتدا ماکزیمم مقدار در هر بردار جواب را به‌عنوان پربارترین ماشین فیزیکی و مینیمم مقدار را به‌عنوان کم‌بارترین ماشین فیزیکی انتخاب می‌نمائیم و ایندکس آن‌ها را ذخیره می‌کنیم.

به‌عنوان مثال بردار جواب (کروموزوم) نمایش داده‌شده در شکل ۵-۴ نشان‌دهنده ۱۰ ماشین فیزیکی در محیط ابری می‌باشد که بنابر فرض مسئله شرایط پذیرش ماشین مجازی را داشته‌اند و برچسب‌های داخل هر سلول، نشان‌دهنده تعداد ماشین مجازی جایابی شده در آن ماشین فیزیکی می‌باشد.

physical machines	7	3	5	2	2	3	1	2	3	2
	PM1	PM2	PM3	PM4	PM5	PM6	PM7	PM8	PM9	PM10

شکل ۳- یک بردار جواب شامل لیست ماشین‌های فیزیکی با جمعیت مختلف

در این بردار جواب، ایندکس ۱ به‌عنوان پربارترین ماشین فیزیکی (ماکزیمم) و ایندکس ۷ به‌عنوان کم‌بارترین ماشین فیزیکی (مینیمم) شناسایی می‌شوند. حال برداری از ماشین‌های مجازی در نظر می‌گیریم، به‌گونه‌ای که برچسب هر کدام نمایشگر شماره ماشین فیزیکی که در آن مستقر هستند می‌باشد.

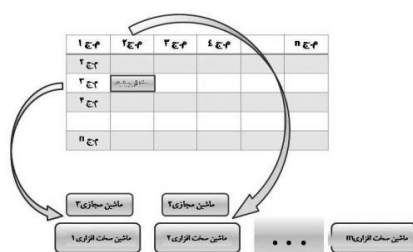
VM1	VM2	VM3	VM4	VM5	VM6	VM7	VM8	VM9	VM30
5	1	2	1	5	1	1	3	10	2

شکل ۴- شاخص ماشین فیزیکی‌ای که هر ماشین مجازی روی آن جایابی شده است

در هر تکرار به بردار ماشین‌های مجازی رفته و به کمک کد متلب^۱ از چپ به راست حرکت می‌کنیم، اولین ماشین مجازی‌ای که در سلول آن ایندکس ماکزیمم باشد را انتخاب کرده و ایندکس آن را با ایندکس مینیمم تعویض می‌نمائیم.

در این مثال اولین ماشین مجازی که ایندکس آن ۱ باشد (یعنی در ماشین فیزیکی اول جایابی شده باشد)، VM2 می‌باشد که در تکرار اول ایندکس آن به ۷ که کم‌بارترین ماشین فیزیکی است، تغییر می‌یابد. با این کار در واقع اولین ماشین مجازی‌ای که روی پربارترین ماشین فیزیکی باشد را به کم‌بارترین ماشین فیزیکی مهاجرت می‌دهیم.

لازم به توضیح است در هر تکرار تنها یک ایندکس تغییر می‌کند، یعنی تنها یک ماشین مجازی مهاجرت می‌یابد، به این ترتیب با هجوم یک‌باره ماشین‌های مجازی به سمت ماشین فیزیکی که شایستگی بیشتری را دارد مواجه نمی‌شویم و در نتیجه میزان تداخل ماشین‌های مجازی حین انتقال صفر می‌شود.



شکل ۵- مهاجرت VM از پربارترین PM به کم‌بارترین PM

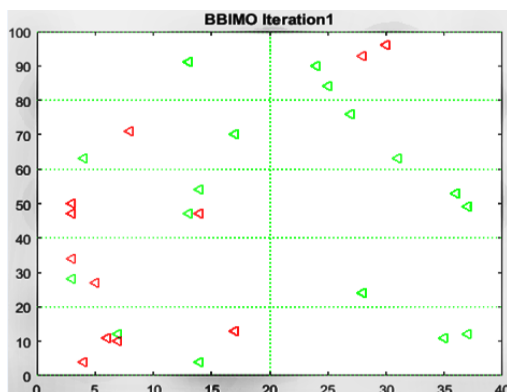
همان‌طور که پیش‌تر توضیح داده شد این الگوریتم یک الگوریتم چند هدفه برای جایابی ماشین‌های مجازی می‌باشد و بر اساس آن می‌توان ادعا نمود بعد از اتمام تکرارهای برنامه، جایابی ماشین‌های مجازی با ایجاد حداکثر توازن بار در نظر گرفتن شایستگی ماشین‌های فیزیکی و تداخل صفر انجام می‌شود.

^۱. Matlab simulation software

۵- ارزیابی الگوریتم BBIMO

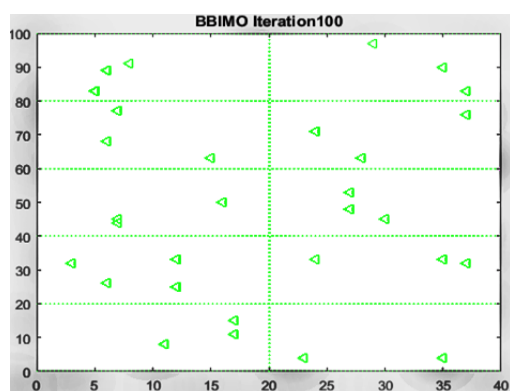
در قسمت بررسی و ارزیابی الگوریتم پیشنهادی، سعی کردیم به کمک الگوریتم BBIMO تعداد ۳۰ ماشین مجازی را بر روی ۱۰ ماشین فیزیکی با شایستگی‌های متفاوت و تعداد ۱۰۰ تکرار جایابی نماییم.

در قسمت ویژوال الگوریتم برای هر ماشین فیزیکی یک کادر مستطیلی در نظر گرفتیم و ماشین‌های مجازی مستقر بر روی آن‌ها را به صورت مثلث‌های سبز و قرمز نمایش دادیم. مثلث‌های سبز نشان‌دهنده ماشین‌های مجازی‌ای است که بر روی ماشین فیزیکی کاملاً مستقر و تثبیت شده‌اند و ماشین‌های مجازی قرمز رنگ، نشان‌دهنده ماشین‌های مجازی است که در تکرار بعدی احتمال انتخاب شدن برای مهاجرت را دارند.



شکل ۶- جایابی ۳۰ ماشین مجازی بر روی ۱۰ ماشین فیزیکی (تکرار اول)

در نهایت بعد از ۱۰۰ تکرار خروجی نهایی به صورت زیر نمایش داده شد، در تکرار آخر تمام ماشین‌های مجازی به نشانه تثبیت شدن، سبز رنگ نمایش داده می‌شوند.



شکل ۷- جایابی ۳۰ ماشین مجازی بر روی ۱۰ ماشین فیزیکی (تکرار صدم)

همان‌طور که در شکل مشهود است در خروجی نهایی ضمن اینکه تعادل بار به خوبی رعایت شده است، با از بین بردن تداخل، جایابی ماشین‌های مجازی را بهبود بخشیده‌ایم.

نتیجه‌گیری

روش‌های موجود برای حل مسئله جایابی ماشین مجازی عمدتاً شامل تعداد زیادی متغیر، فرمول و محدودیت می‌باشند که از کارایی اولیه آن‌ها در حل مسائل با ابعاد واقعی می‌کاهد و همچنین انرژی زیادی صرف عملیات انتخاب ماشین مقصد و انتقال ماشین مجازی به آن می‌شود، از این رو استفاده از روش‌ها و الگوریتم‌های ابتکاری و فرا ابتکاری در حل این مسائل کاملاً به‌جا می‌باشد.

رویکرد پیشنهادی در راستای رسیدن به یک تخصیص مناسب با حداقل زمان اجرا و حداقل برخورد ماشین‌های مجازی با یکدیگر ارائه شده است. همان‌طور که گفته شد برای بهبود جایابی ماشین‌های مجازی از الگوریتم بهینه‌سازی مبتنی بر جغرافیای زیستی استفاده کردیم و سعی کردیم با در نظر گرفتن دو آیتم مهاجرت‌پذیری و مهاجرت‌گریزی بهترین مقصد را برای هر ماشین مجازی در حال انتقال انتخاب نماییم.

در نهایت بردارهای جواب را در هر مرحله به‌گونه‌ای بهبود بخشیدیم که در هر واحد زمانی یک ماشین مجازی از پربارترین ماشین فیزیکی به کم‌بارترین ماشین فیزیکی مهاجرت یابد. به این ترتیب علاوه بر از بین بردن تداخل ماشین‌های مجازی در حین انتقال، شاهد افزایش تعادل بار و کاهش زمان اجرا در جایابی ماشین‌های مجازی خواهیم بود.

منابع و مراجع

- [1] D. Simon, "Biogeography-based optimization", IEEE Trans. on Evolutionary Computation. Vol. 12, No. 6, pp.702-713, 2008
- [2] M.F.Gholamia,F.Daneshgara, G.Lowa, Gha.Beydounc," Cloud migration process—A survey, evaluation framework, and open challenges", Elsevier, 2016
- [3] Q.Zheng, J.Li, B.Dong, R.Li, N.Shah, F.Tian, " Multi-objective Optimization Algorithm based on BBO for Virtual Machine Consolidation Problem",IEEE, 2015